Australian Government
Department of Defence
Defence Science and
Technology Organisation

Trim Calculation Methods for a Dynamical Model of the REMUS 100 Autonomous Underwater Vehicle

Raewyn Hall and Stuart Anstee

Maritime Operations Division
Defence Science and Technology Organisation

DSTO-TR-2576

ABSTRACT

The calculations given in this work demonstrate that the trimmed state for a dynamical model of the REMUS 100 autonomous underwater vehicle is readily found using a numerical zero-finding procedure based on Newton-Raphson iteration. This work also presents approximate analytical expressions for the trimmed state that can be used as a starting point for the numerical procedure. The procedure should be applicable to a range of hydrodynamic parameters corresponding to other configurations of the REMUS 100 vehicle and to similar vehicles from other vendors.

RELEASE LIMITATION

Approved for public release

Published by

Maritime Operations Division
DSTO Defence Science and Technology Organisation
PO Box 1500
Edinburgh South Australia 5111 Australia

Telephone: (08) 7389 5555
Fax: (08) 7389 6567

AR-015-046
August 2011

APPROVED FOR PUBLIC RELEASE

Trim Calculation Methods for a Dynamical Model of the REMUS 100 Autonomous Underwater Vehicle

Executive Summary

Survey-class autonomous underwater vehicles (AUVs) are used to investigate the seabed and the water column with high resolution and navigational accuracy. The Royal Australian Navy (RAN) is in the process of acquiring AUVs for use in mine countermeasures, Rapid Environmental Assessment and Advance Force operations. AUVs are potentially major components of acquisition projects SEA 1778 Phase 1 *Deployable MCM – Organic Mine Countermeasures* and JP 1770 Phase 1 *Rapid Environmental Assessment*. They may also be components of off-board mission systems that will be acquired as part of future projects SEA 1180 Phase 1 - *Patrol Boat, Mine Hunter Coastal and Hydrographic Ship Replacement Project* and SEA 1000, *Future Submarine*.

The DSTO acquired two commercial AUVs in 2007, in collaboration with the Directorate General of Maritime Development and the RAN. One of the AUVs was a REMUS 100, manufactured by Kongsberg Hydroid Incorporated of the USA. Many navies have one or more REMUS 100 vehicles in their inventory and the number of REMUS 100s that has been produced is of the same order as the number of all other commercial AUVs in existence, combined. Although the DSTO vehicle has been extensively tested, its high value precludes testing in waters where currents are strong, or shallow and wave-driven, since the vehicle might be damaged, lost or destroyed. However, operating areas in which such conditions occur are potentially of high military value and there is considerable interest in being able to predict the limiting conditions for use of the vehicle. As a consequence, the DSTO has begun to investigate techniques by which the dynamics of the vehicle may be simulated with appropriate fidelity, using so-called 'low-level' simulation models based on estimates of the governing equations of motion for the vehicle.

Accurate low-level simulations of vehicle behaviour rely on well-behaved numerical implementations. One component of such implementations is the 'trim' state of the model, which, for a given speed, is the combination of vehicle orientation and control settings in which the unperturbed vehicle will maintain straight-line, level flight in a state of dynamic equilibrium. The trim state is used to analyse the stability of the numerical model – its response to perturbations away from the trim state – and to initialise simulations.

This work describes a robust, parametric process to estimate the trim state of a dynamical model of the REMUS 100 AUV based on equations of motion originally developed by Prestero [1] and extended by Sgarioto [2]. The method is based on direct solution of the governing equations of motion, using an iterative Newton-Raphson zero-finding algorithm. The algorithm is initialised using second-order analytical approximations to the

Prestero-Sgarioto equations of motion. The relative accuracy of the analytical expressions is better than 1% at speeds near the cruising speed of the vehicle, but decreases by 1 to 2 orders of magnitude at lower speeds. Other methods of initialisation were also investigated and a reduced set of analytical expressions was found to be effective as a starting point for the iterative calculation.

A parametric investigation showed that the iterative algorithm was convergent for speeds from 1 knot to at least 6 knots; in comparison, the physical vehicle operates within a range of speeds between 2 knots and 5 knots. The variation of the trimmed state conformed to expectations over the latter range. Unexpected behaviour was seen at lower speeds, but this did not originate from a failure of the trim calculation.

In summary, this work has resulted in a procedure for finding the trimmed state of the Prestero AUV dynamics model using a straightforward numerical method with a parametric initialisation procedure. With appropriate parameters, the method should be applicable to variations of the REMUS vehicle; for example, extended versions, and to other REMUS-like vehicles.

References cited in this section:

1. Prestero, T. “Verification of a Six-Degree-of-Freedom Simulation Model for the REMUS Autonomous Underwater Vehicle”, MSc/ME Thesis, Massachusetts Institute of Technology, Sept. 2001.

2. Sgarioto, D. “Steady State Trim and Open Loop Stability Analysis for the REMUS Autonomous Underwater Vehicle”, Defence Technology Agency, New Zealand Defence Force, DTA Report 254, March 2008.

Authors

Raewyn Hall
Maritime Operations Division

Raewyn graduated from Sydney University in 2005 with honours in Aeronautical (Space) Engineering and a Bachelor of Science. Her engineering honours thesis focussed on aircraft flight dynamics, guidance and model predictive control. She joined the Maritime Operations Division in 2006 doing operations analysis and modelling in a variety of areas such as amphibious operations and anti-submarine warfare. She currently splits her time between the Maritime Concepts and Capabilities Group and modelling AUV dynamics for the Littoral Unmanned Systems Group.

Stuart Anstee
Maritime Operations Division

Stuart Anstee is a member of the Littoral Unmanned Systems Group, which investigates the application of unmanned vehicles to mine warfare and hydrography. In his career at DSTO, he has worked on the design, modelling and assessment of high-frequency sonars, analysis tools for sonar imagery and operations research. His current interests include assessment of autonomous vehicle systems, the hydrodynamics and control of underwater vehicles, autonomous mission planning and investigation of sensors for mine warfare and hydrography.

This page is intentionally blank

Contents

1. INTRODUCTION 1

2. THE REMUS 100 MODEL 1
2.1 The REMUS 100 Vehicle 1
2.2 Simulation Model 2
2.2.1 Notation 2
2.2.2 Mathematical Representation 4
2.3 Coordinate Transformations 4
2.3.1 Earth and Body Reference Frames 4
2.3.2 Body and Stability Reference Frames 5
2.4 Equations of Motion 6
2.5 External Forces and Moments 7

3. ESTIMATION OF TRIM 8
3.1 Definition of Trim 8
3.2 Trim Variables 8
3.3 Decision Variables 9
3.4 Trim Estimation by Numerical Iteration 9
3.4.1 Newton-Raphson Iteration 9
3.4.2 Implementation 10
3.4.3 Performance of the Newton-Raphson Iteration 10
3.4.4 Convergence Sensitivity 11
3.5 Estimation of the Starting Point for the Iteration 12
3.5.1 Empirical and Analytical Starting Points 12
3.5.2 Simplified Analytical Starting Point 13
3.5.3 Results and Discussion 13

4. PARAMETRIC STUDY 14
4.1 Vehicle Orientation Angles 14
4.2 Propulsion Variables 15
4.3 Control Inputs 16

5. CONCLUSION 17

6. REFERENCES 18

APPENDIX A: REMUS 100 MODEL PARAMETERS 19
A.1. Mathematical Symbols 19
A.2. Case Study Parameters 20

APPENDIX B: ALGORITHM IMPLEMENTATION 23

B.1. Interface Function 23
B.2. Iteration Algorithm 24
B.2.1 Jacobian matrix calculation 24
B.3. Trim function MATLAB script 26
B.3.1 Main Trim Function *'trim.m'* 26
B.3.2 Equations of Motion Function *'eom.m'* 28
B.3.3 Hydrodynamics Coefficients script file *'coeffs.m'* 31

APPENDIX C: ANALYTICAL TRIM ESTIMATION USING FORCE BALANCE CONDITIONS 34
C.1. Senses of Motion 34
C.2. Approximation to Nearly Level Motion 35
C.3. Derivation of Analytical Approximations 37
C.3.1 Propulsion Subsystem 37
C.3.2 Manoeuvre Subsystem 37
C.4. Comparison of Numerical and Analytical Trim Estimates 38

APPENDIX D: CONVERGENCE SENSITIVITY ANALYSIS 43

GLOSSARY

AUV	Autonomous Underwater Vehicle
GPS	Global Positioning System
INS	Inertial Navigation System
LBL	Acoustic Long Baseline navigation system
RAN	Royal Australian Navy

This page is intentionally blank

1. Introduction

Survey-class autonomous underwater vehicles (AUVs) are used to investigate the seabed and the water column with high resolution and navigational accuracy. The Royal Australian Navy (RAN) is in the process of acquiring AUVs for use in mine countermeasures, Rapid Environmental Assessment (REA) and Advance Force operations. AUVs are potentially major components of acquisition projects SEA 1778 Phase 1 *Deployable MCM – Organic Mine Countermeasures* and JP 1770 Phase 1 *Rapid Environmental Assessment*. They may also be components of off-board mission systems that will be acquired as part of future project SEA 1180 Phase 1 - *Patrol Boat, Mine Hunter Coastal and Hydrographic Ship Replacement Project* and SEA 1000, *Future Submarine*.

The DSTO acquired two commercial AUVs in 2007, in collaboration with the Directorate General of Maritime Development and the RAN. One of the AUVs was a REMUS 100, manufactured by Kongsberg Hydroid Incorporated of the USA. Although this vehicle has been extensively tested, its high value precludes its inclusion in trials where currents are excessively strong or waters are shallow and wave-driven, since damage or loss of the vehicle might result. As a consequence, the DSTO has begun to investigate techniques by which the dynamics of the vehicle might be simulated in extreme environments. High-fidelity simulations of vehicle behaviour rely on well-behaved numerical models. A starting point for such models is the 'trim' state of the vehicle, corresponding to the control state in which it will maintain straight-line, level flight in a state of dynamic equilibrium.

This report describes analytical and numerical methods that may be used to trim a non-linear, six degree-of-freedom simulation model describing the dynamics of a REMUS 100 AUV. The convergence properties of the trim calculation and the nature of the trim state are also examined as a function of the model parameters.

2. The REMUS 100 Model

2.1 The REMUS 100 Vehicle

The REMUS 100 is a slender, torpedo-shaped vehicle, as shown in Figure 1. It is approximately 1.6 to 1.72 m long, has a diameter of 19 cm and weighs 36 to 40 kg in air, depending on configuration. It is designed to execute pre-programmed missions in order to collect sonar imagery of the seafloor, operating at ground speeds from 2.5 to 4.5 knots.

The main payload sensor is side-scan sonar, which is typically used to gather imagery of the sea floor. The vehicle has a suite of other sensors, including a pressure sensor, a conductivity sensor, a thermometer, an acoustic long-baseline (LBL) transceiver, upward and downward-pointing acoustic Doppler current profilers and a GPS receiver, all of which may optionally be used to aid an inertial navigation system (INS).

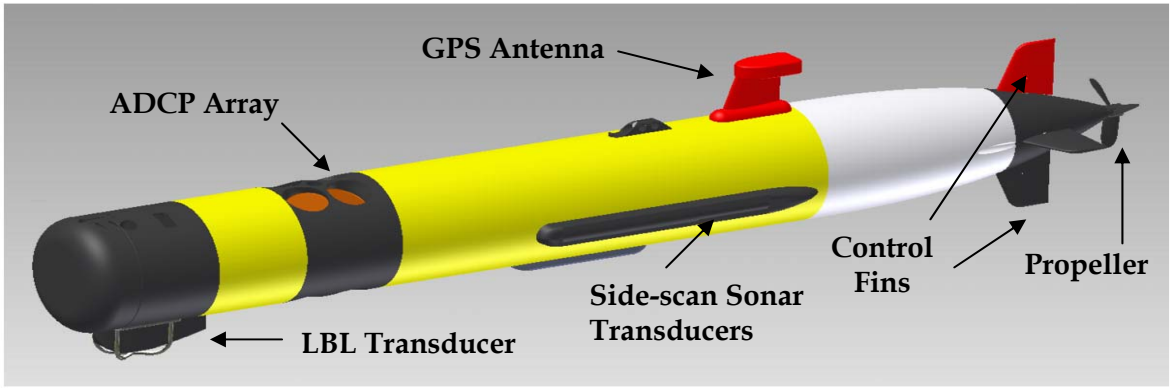

Figure 1: One configuration of the REMUS 100 vehicle

The vehicle is under-actuated, with control provided by paired fins and a motor-driven propeller. Fused[1] horizontal fins control the vehicle's pitching motion and fused vertical fins control the vehicle's yawing motion. The control system also continuously varies the torque and rate of rotation of the coreless electric motor that drives the propeller to control the vehicle's forward thrust and hence its speed. The control system alternates pitching and yawing motions to control rolling motions induced by the torque from the propeller.

2.2 Simulation Model

This work is based on a six degree-of-freedom, medium-fidelity model of the vehicle dynamics that was originally developed by Prestero [1] and extended by Sgarioto [4]. The model implements slender body equations of motion, with the variation of external forces acting on the vehicle determined by hydrodynamic coefficients. The model equations are given in this section and the coefficients are recorded in Appendix A.

2.2.1 Notation

The notations used in this report to describe rigid-body dynamics follow the conventions of the Society of Naval Architects and Marine Engineers (SNAME) used in [1] and [2] and reproduced in Table 1. Euler angles (ϕ, θ, ψ) describe the orientation of the body-fixed axes (x, y, z) relative to some Earth-fixed reference frame (X, Y, Z).

The sense of the axes is right-handed; hence x is positive forward, y is positive to starboard, z is positive down, θ is positive nose-up, ϕ is positive clockwise and ψ is positive to starboard.

[1] Fused in this context means 'forced to move in unison'

Table 1: Symbols used to describe rigid-body dynamics in the body-fixed reference frame

Degree of Freedom	External Forces and Moments	Corresponding Rates of Change	Corresponding Displacements
Surge	X	U	x
Sway	Y	V	y
Heave	Z	W	z
Roll	K	P	ϕ
Pitch	M	Q	θ
Yaw	N	R	ψ

The variables and their respective axes are shown in Figure 2. The axes of the Earth-fixed reference frame (X, Y, Z) are displayed with a different font to that used later for the components of linear external force X, Y, Z.

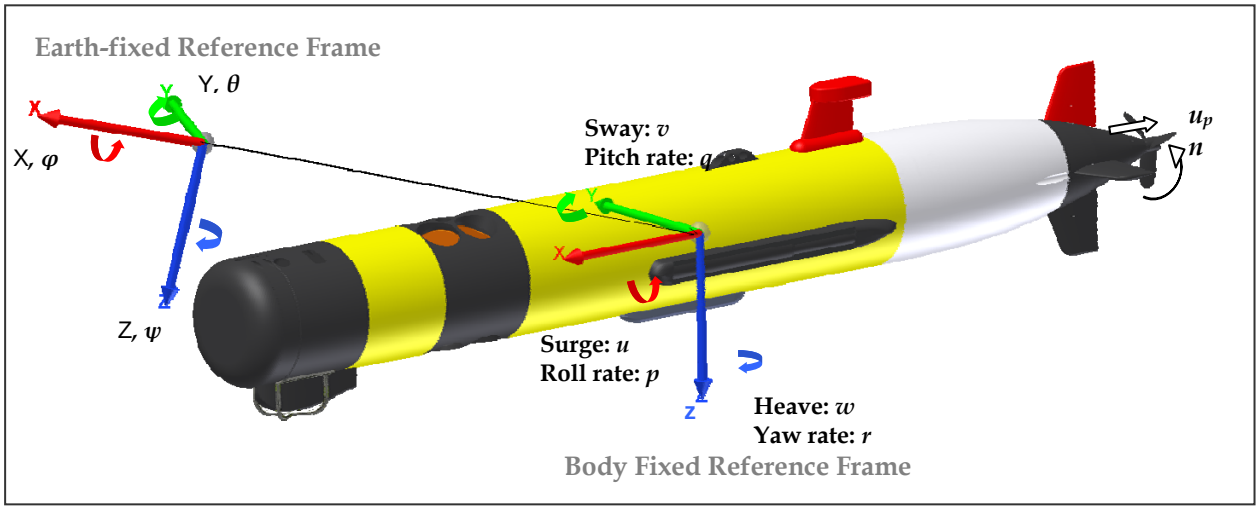

Figure 2: REMUS simulation model frames of reference. The position and orientation of the vehicle are described with reference to an Earth-fixed frame, while rates of translation and rotation are expressed relative to a frame attached to the body of the vehicle.

Additional symbols describing the state of the propulsion system are given in Table 2.

Table 2: Symbols used to describe the state of the propulsion system

Name	Symbol
Propeller rotation rate	N
Propeller inflow velocity	u_p
Control Torque	τ

2.2.2 Mathematical Representation

The vehicle's 'state' at any instant in time is described mathematically by a state vector, incorporating its rates of translation and rotation, its absolute orientation and position, and the status of its propulsion system:

$$\boldsymbol{\Xi} = [u \quad v \quad w \quad p \quad q \quad r \quad \phi \quad \theta \quad \psi \quad \mathsf{X} \quad \mathsf{Y} \quad \mathsf{Z} \quad n \quad u_p]^T, \tag{2.2.1a}$$

where 'T' denotes the transpose operator.

The velocity components of the vehicle can also be described relative to the water flow as (V, a, β) (see Section 2.3.2), giving rise to the alternative state vector

$$\boldsymbol{\Xi} = [V \quad \alpha \quad \beta \quad p \quad q \quad r \quad \phi \quad \theta \quad \psi \quad \mathsf{X} \quad \mathsf{Y} \quad \mathsf{Z} \quad n \quad u_p]^T. \tag{2.2.1b}$$

The vehicle's control state is likewise described by a control input vector,

$$\mathbf{u} = \left[\delta_s \quad \delta_r \quad \tau\right]^T. \tag{2.2.2}$$

This vector comprises the elevator angle δ_s, the rudder angle δ_r and the mechanical torque the motor exerts around the propeller axis, τ.

The mathematical model of the vehicle is implemented in the form of a function which takes the vehicle state and the control input in the form of vectors $\boldsymbol{\Xi}$ and $\mathbf{u}$ and predicts the rate of the change of each of the vehicle state variables in the form

$$\dot{\boldsymbol{\Xi}}(t) = f\left(\boldsymbol{\Xi}(t), \mathbf{u}(t), t\right). \tag{2.2.3}$$

The components of the rate of change equations are given in Section 2.4.

2.3 Coordinate Transformations

2.3.1 Earth and Body Reference Frames

The equations in Section 2.4 describe the vehicle state in the body-fixed frame of reference. Thus, transformations are necessary to move between body-fixed and earth-fixed coordinate systems. The linear velocity components $(\dot{\mathsf{X}}, \dot{\mathsf{Y}}, \dot{\mathsf{Z}})$ in the earth-fixed reference frame are related to the body-frame velocity components (u, v, w) by:

$$\begin{aligned}\dot{\mathsf{X}} = {} & u\cos\psi\cos\theta + v(-\sin\psi\cos\phi + \cos\psi\sin\theta\sin\phi) \\ & + w(\sin\psi\sin\phi + \cos\psi\sin\theta\cos\phi)\end{aligned} \tag{2.3.1}$$

$$\dot{Y} = u\sin\psi\cos\theta + v(\cos\psi\cos\phi + \sin\psi\sin\theta\sin\phi) + w(-\cos\psi\sin\phi + \sin\psi\sin\theta\cos\phi) \quad (2.3.2)$$

$$\dot{Z} = -u\sin\theta + v\cos\theta\sin\phi + w\cos\theta\cos\phi \quad (2.3.3)$$

The rate equations for the earth-fixed Euler angles (ϕ, θ, ψ) are similarly expressed in terms of the body-fixed rates of turn (p, q, r) by:

$$\dot{\phi} = p + q\sin\phi\tan\theta + r\cos\phi\tan\theta \quad (2.3.4)$$

$$\dot{\theta} = q\cos\phi - r\sin\phi \quad (2.3.5)$$

$$\dot{\psi} = q\sin\phi/\cos\theta + r\cos\phi/\cos\theta \quad (2.3.6)$$

2.3.2 Body and Stability Reference Frames

For convenience of interpretation, the vehicle state is projected into the so-called 'stability' reference frame; that is, the reference frame attached to the body of water surrounding the vehicle, which in the absence of current is identical to the earth-fixed reference frame. In this case, the body-axis velocity components (u, v, w) are replaced by the forward speed, angle of attack and sideslip angle (V, α, β), defined by

$$V = \sqrt{u^2 + v^2 + w^2} \quad (2.3.7)$$

$$\alpha = \tan^{-1}(w/u) \quad (2.3.8)$$

$$\beta = \sin^{-1}(v/V) \quad (2.3.9)$$

Equivalently,

$$u = V\cos\alpha\cos\beta \quad (2.3.10)$$

$$v = V\sin\beta \quad (2.3.11)$$

$$w = V\sin\alpha\cos\beta \quad (2.3.12)$$

The rates of change of the stability variables are given in terms of the rates of change of quantities expressed in the body-frame by the following equations:

$$\dot{V} = \dot{u}\cos\alpha\cos\beta + \dot{v}\sin\beta + \dot{w}\sin\alpha\cos\beta \quad (2.3.13)$$

$$\dot{\alpha} = \frac{\dot{w}\cos\alpha - \dot{u}\sin\alpha}{V\cos\beta} \quad (2.3.14)$$

$$\dot{\beta} = \frac{1}{V}\left(\dot{v}\cos\beta - \dot{u}\cos\alpha\sin\beta - \dot{w}\sin\alpha\sin\beta\right) \quad (2.3.15)$$

2.4 Equations of Motion

The following equations describe the motion of the vehicle and the behaviour of its propulsion system in a body-fixed frame of reference.

The equations of motion for quantities driven by external forces are:

$$m\left[\dot{u} - vr + wq + z_g\left(pr + \dot{q}\right)\right] = \sum X \tag{2.4.1}$$

$$m\left[\dot{v} - wp + ur + z_g\left(qr - \dot{p}\right)\right] = \sum Y \tag{2.4.2}$$

$$m\left[\dot{w} - uq + vp - z_g\left(p^2 + q^2\right)\right] = \sum Z \tag{2.4.3}$$

The quantity z_g is the vertical offset between the centre of buoyancy and the centre of mass in the body frame of reference. It is typically positive, indicating that the centre of gravity is below the centre of buoyancy.

The components $\sum X, \sum Y$ & $\sum Z$ of the external linear forces are given in Section 2.5.

The equations of motion for quantities driven by external moments are:

$$I_{yy}\dot{q} + \left(I_{xx} - I_{zz}\right)rp + mz_g\left(\dot{u} - vr + wq\right) = \sum M \tag{2.4.4}$$

$$I_{xx}\dot{p} + \left(I_{zz} - I_{yy}\right)qr - mz_g\left(\dot{v} - wp + ur\right) = \sum K \tag{2.4.5}$$

$$I_{zz}\dot{r} + \left(I_{yy} - I_{xx}\right)pq = \sum N \tag{2.4.6}$$

The components $\sum K, \sum M$ & $\sum N$ of the external moments are given in Section 2.5.

The propulsion system equations are:

$$J_m\dot{n} = Q - K_n n - Q_{n|n|}n|n| \tag{2.4.7}$$

$$m_f\dot{u}_p = T_{n|n|}n|n| - d_{f_0}u_p - d_f\left|u_p\right|\left(u_p - (1 - w_p)u\right) \tag{2.4.8}$$

where d_{f_0} & d_f are damping coefficients defined by:

$$d_{f_0} = \frac{-2X_{u|u|}}{(1-\tau_p)(1+a_p)(1-w_p)}$$

$$d_f = \frac{-X_{u|u|}}{(1-\tau_p)(1+a_p)a_p(1-w_p)^2}$$

Refer to Table 9 of Appendix A.2 for the values of these parameters.

2.5 External Forces and Moments

The external forces acting on the rigid body of the vehicle are composed of hydrostatic forces, hydrodynamic forces and forces due to the control surfaces and propeller; that is,

$$\sum F_{ext} = F_{hydrostatic} + F_{lift} + F_{drag} + F_{control} . \tag{2.5.1}$$

After expansion, the individual components of the external forces obey the following equations:

$$\begin{aligned}\Sigma X = & -(W-B)\sin\theta + X_{u|u|}u|u| + X_{\dot{u}}\dot{u} + X_{wq}wq + X_{qq}q^2 + X_{vr}vr + X_{rr}r^2 \\ & +(1-\tau_p)T_{n|n|}n|n|\end{aligned} \tag{2.5.2}$$

$$\begin{aligned}\Sigma Y = & (W-B)\cos\theta\sin\phi + Y_{uv}uv + Y_{v|v|}v|v| + Y_{r|r|}r|r| + Y_{\dot{v}}\dot{v} + Y_{\dot{r}}\dot{r} + Y_{ur}ur \\ & +Y_{wp}wp + Y_{pq}pq + Y_{uu\delta_r}u^2\delta_r\end{aligned} \tag{2.5.3}$$

$$\begin{aligned}\Sigma Z = & (W-B)\cos\theta\cos\phi + Z_{\dot{w}}\dot{w} + Z_{\dot{q}}\dot{q} + Z_{uq}uq + Z_{vp}vp + Z_{rp}rp + Z_{w|w|}w|w| \\ & + Z_{q|q|}q|q| + Z_{uw}uw + Z_{uu\delta_s}u^2\delta_s .\end{aligned} \tag{2.5.4}$$

Similarly, the moment expansions are:

$$\Sigma K = -Wz_g\cos\theta\sin\phi + K_{\dot{p}}\dot{p} + K_{p|p|}p|p| + Q_{n|n|}n|n| \tag{2.5.5}$$

$$\begin{aligned}\Sigma M = & -Wz_g\sin\theta + M_{\dot{w}}\dot{w} + M_{\dot{q}}\dot{q} + M_{uw}uw + M_{vp}vp + M_{rp}rp + M_{uq}uq \\ & +M_{w|w|}w|w| + M_{q|q|}q|q| + M_{uu\delta_s}u^2\delta_s\end{aligned} \tag{2.5.6}$$

$$\begin{aligned}\Sigma N = & N_{\dot{v}}\dot{v} + N_{\dot{r}}\dot{r} + N_{uv}uv + N_{wp}wp + N_{pq}pq + N_{ur}ur + N_{v|v|}v|v| \\ & + N_{r|r|}r|r| + N_{uu\delta_r}u^2\delta_r .\end{aligned} \tag{2.5.7}$$

Refer to Appendix A for definitions of the coefficients and their values.

After substituting the force and moment expressions (2.5.2) to (2.5.7) into the equations of motion (2.4.1) to (2.4.8), we obtain the full set of non-linear equations expressed as a function of the vehicle states and state-rates. It is then possible to recast the equations into state-space form. This is the form in which it is easiest to simulate the motion of the vehicle and calculate its trim state.

3. Estimation of Trim

3.1 Definition of Trim

A trimmed vehicle is one that, when unperturbed, maintains a state of dynamic equilibrium in level, straight-line motion; that is, if it experiences no perturbations, it will maintain a fixed velocity and orientation, without requiring adjustments to the control input settings. Finding the trim state of a body is a complex problem, due to the variety of external forces and moments acting upon it; they are illustrated in Figure 3.

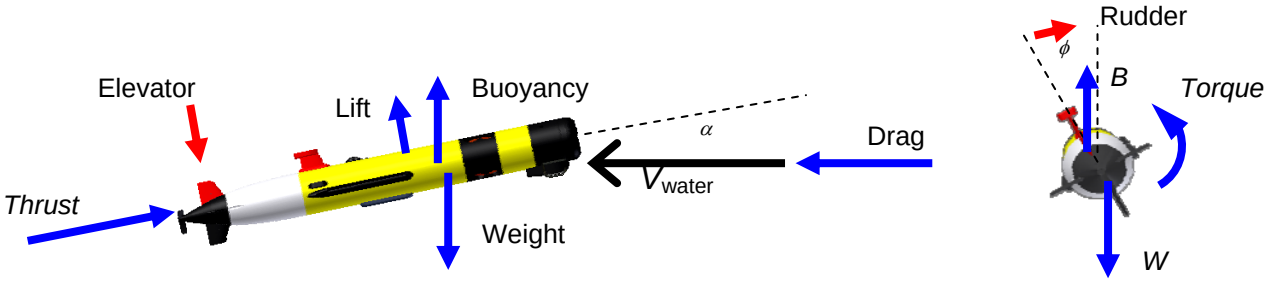

Figure 3: A representation of the external forces and moments acting on a REMUS AUV

In order to maintain a constant speed, depth and heading, all the forces and moments acting about all axes must be balanced, that is, they must sum to zero. The vehicle must also have zero rates of rotation about all three axes.

3.2 Trim Variables

The independent variable in the trim calculation is chosen to be the speed of the vehicle through the water, V. The trim calculation consists of estimating the control parameters - elevator angle, rudder angle and propeller torque - which result in a vehicle orientation in which all the forces and moments acting on the vehicle are balanced.

In dynamic equilibrium, the rates of change of all components of the state vector $\boldsymbol{\Xi}$ except horizontal position are zero; that is, all components of $\dot{\boldsymbol{\Xi}}$ other than $\dot{X}$ and $\dot{Y}$ are zero. We refer to the remaining components of $\dot{\boldsymbol{\Xi}}$ by a vector $\boldsymbol{\Omega}$ of rates of change given by

$$\boldsymbol{\Omega} = [\dot{V}, \dot{\alpha}, \dot{\beta}, \dot{p}, \dot{q}, \dot{r}, \dot{Z}, \dot{\phi}, \dot{\theta}, \dot{\psi}, \dot{n}, \dot{u}_p]^T.$$

Note that the condition $\dot{\phi} = \dot{\theta} = \dot{\psi} = 0$ implies that the vehicle is not turning so the body-frame rotation rates are zero, that is, $p = q = r = 0$.

The object of the trim calculation is to drive the components of $\boldsymbol{\Omega}$ to zero.

3.3 Decision Variables

A set of decision variables must be chosen to facilitate the trim calculation. In this case, for a required forward speed V, the variables chosen are

$$\mathbf{x} = [\alpha, \beta, \phi, \theta, n, u_p, \tau, \delta_s, \delta_r]^T .$$

This set does not include the linear velocities (u, v, w), because the information they represent is present in the values of (V, a, β); inclusion of (u, v, w) results in an over-determined system.

3.4 Trim Estimation by Numerical Iteration

The trim calculation is the problem of finding those values of the decision variables that drive the absolute value of the trim variables to zero. One approach to this problem is to use numerical iteration based on the equations of motion.

3.4.1 Newton-Raphson Iteration

If the individual components of $\mathbf{\Omega}$ are differentiable and monotonic in the neighbourhood of the trim point, we can proceed via Newton-Raphson iteration. In the multivariate case, the calculation is initialised with some reasonable decision vector value $\mathbf{x}_0$ and the iteration proceeds as

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{J}^{-1}\mathbf{\Omega}(\mathbf{x}_k) \tag{3.4.1}$$

where $\mathbf{J}$ is the Jacobian matrix

$$\mathbf{J} = \begin{bmatrix} \frac{\partial \Omega_1}{\partial x_1} & \frac{\partial \Omega_1}{\partial x_2} & \cdots & \frac{\partial \Omega_1}{\partial x_n} \\ \frac{\partial \Omega_2}{\partial x_1} & \frac{\partial \Omega_2}{\partial x_2} & & \frac{\partial \Omega_2}{\partial x_n} \\ \vdots & & \ddots & \vdots \\ \frac{\partial \Omega_m}{\partial x_1} & \frac{\partial \Omega_m}{\partial x_2} & \cdots & \frac{\partial \Omega_m}{\partial x_n} \end{bmatrix}. \tag{3.4.2}$$

Although analytical calculation of the elements of the Jacobian matrix is possible, doing so is laborious; in this work, individual elements of the matrix were estimated using the first-order approximation

$$\frac{\partial \Omega_i}{\partial x_j} = \frac{1}{\Delta x}\left(\mathbf{\Omega}(x_1,\ldots,x_j+\Delta x,\ldots,x_n) - \mathbf{\Omega}(x_1,\ldots,x_j,\ldots,x_n)\right). \tag{3.4.3}$$

The convergence of the algorithm is measured by comparing the difference between the current trim vector and the previous trim vector, referred to as the 'error', ε

$$\varepsilon = \sum_k \left| \mathbf{x}_{k+1} - \mathbf{x}_k \right| \tag{3.4.4}$$

The iteration was terminated when error was less than a desired tolerance value, ε_{tol},

$$\varepsilon < \varepsilon_{tol} . \tag{3.4.5}$$

In the following sections of this work, the first-order approximation to the partial derivatives in the Jacobian matrix was calculated with a uniform perturbation value of

$$\Delta x = 0.001. \tag{3.4.6}$$

The iteration was terminated when the absolute difference between successive estimates fell below

$$\varepsilon_{tol} = 10^{-10}. \tag{3.4.7}$$

Alternatively, the iteration was deemed to have failed to converge if the number of iterations reached 100.

3.4.2 Implementation

The trim method described above was implemented as an algorithm in MATLAB code as the function '*trim.m*'. It takes the following inputs:-

- Desired trim speed V and
- Initial trim state estimate $\mathbf{x}_0$ (optional)

The function outputs are:-

- The trim state vector $\mathbf{x}$,
- The number of iterations required to reach to error tolerance, and
- The final error ε.

The trim function calls a function '*eom.m*' that implements the state-space form of the equations of motion. The implementation is detailed in Appendix B.

3.4.3 Performance of the Newton-Raphson Iteration

As a case study, Sgarioto [2] reports the trim state for of the REMUS 100 model using parameters given in Appendix A, at the representative forward speed value of V = 4 knots. The values derived by Sgarioto were calculated using a commercial optimisation software package called SNOPT. For comparison, the Newton-Raphson iteration algorithm described in Section 3.4.1 was initialised using the starting value given by Sgarioto [2]:

$$\mathbf{x}_0 = [\alpha_0, \beta_0, \varphi_0, \theta_0, n_0, u_{p0}, \tau_0, \delta_{s0}, \delta_{r0}]^T = [0, 0, 0, 0, 76V, 0.9V, 39V, 0, 0]. \quad (3.4.8)$$

The results from both processes are shown in Table 3. The agreement is sufficient to imply that the Newton-Raphson iteration described in the previous section is effective.

Table 3: Comparison of the calculations of Sgarioto [2] and this work for a speed V = 4 knots

Variable	Symbol	SNOPT	Newton-Raphson
Body-frame surge velocity	U	2.0575 ms^{-1}	2.0577 ms^{-1}
Body-frame sway velocity	V	0.001 ms^{-1}	0.0010 ms^{-1}
Body-frame heave velocity	W	-0.021 ms^{-1}	-0.0209 ms^{-1}
Earth-frame roll	ϕ	-2.42°	-2.4178°
Earth-frame pitch	θ	-0.583°	-0.5833°
Propeller rotation rate	N	1418 RPM	1418 RPM
Propeller inflow rate	u_p	1.42 ms^{-1}	1.421 ms^{-1}
Angle of attack	α	-0.58°	-0.5826°
Angle of sideslip	β	0.028°	0.0285°
Control torque	τ	74 Nm	74.01 Nm
Elevator angle	δ_s	-2.15°	-2.145°
Rudder angle	δ_r	-0.11°	-0.111°

3.4.4 Convergence Sensitivity

In order to test the robustness of the algorithm, a sensitivity analysis of the trim convergence behaviour was conducted with respect to the hydrodynamic coefficients. As there are over 50 hydrodynamic coefficients in the REMUS 100 model, it was not possible to report on the sensitivity of the convergence to all the parameters. Instead, five coefficients that Sgarioto identified as having significant effects on the longitudinal (x-z plane) stability of the REMUS model [2] were considered: M_{uw}, $M_{uu\delta_s}$, $N_{uu\delta_r}$, $X_{u|u|}$ and Z_{uw}. These coefficients depend on the physical configuration of the vehicle, particularly its length.

Appendix D describes a parametric analysis on the effect of the individual variation of each of the coefficients. This was done by running the trim algorithm for a forward speed of 4 knots with the value of each coefficient scaled to 50%, 75%, 100%, 125% and 150% of its original value, as stated in Appendix A.1.

In summary, the results show that the algorithm converged to a trim condition for all of the modified coefficient values. The number of iterations did not change for the majority of the runs, although in a few cases, the trim algorithm converged slightly faster than it did with the original coefficient values. This suggests that the algorithm should be robust enough to trim different REMUS 100 configurations and other similar AUV models.

3.5 Estimation of the Starting Point for the Iteration

The convergence of the Newton-Raphson zero-finding algorithm expressed by equation (3.4.1) can only be guaranteed for a restricted class of functions and starting points for the iteration [3]. For example, the algorithm may fail to find the global minimum if the starting point is close to a local minimum. In the present case, the existence of local minima is difficult to predict, particularly because the parameters of the model depend on the configuration of the vehicle being modeled. This suggests that it is important to select an appropriate starting point for the iteration in order to guarantee convergence to the global minimum. This section considers methods by which a starting point for the iteration can be derived.

3.5.1 Empirical and Analytical Starting Points

Sgarioto [2] derived equation (3.4.8) for the starting point of the numerical solution of the trim equations empirically, by a process of trial and error that was based on repeatedly integrating the non-linear equations of motion described in Section 2.4 and observing the dynamics of the vehicle as they evolved over long spans of time. Equation (3.4.8) is an appropriate starting point for the Newton-Raphson iteration for forward speeds from 1 knot to 6 knots, with convergence properties shown as the 'Empirical' curves in Figure 4.

Although it was successful in this case, equation (3.4.8) was derived for a particular set of model parameters and it may not be appropriate for different values of the model parameters.

It is possible to derive more widely applicable initial conditions from first principles. By applying a set of steady, level flight assumptions to the vehicle force balance equations and approximating the equations by neglecting terms smaller than second order in small quantities, it is possible to obtain analytical expressions for the trim state of the vehicle. The derivation and form of these analytical expressions are described in Appendix C.

Although the analytical approximations of the trimmed state given as expressions (C.2.15) to (C.3.8) in Appendix C are not sufficiently accurate for use in trim-based calculations over the full range of vehicle speeds[2], they are appropriate starting points for iterative refinement. The convergence properties that result from using these expressions as the starting points for the Newton-Raphson iteration are shown as the 'Analytical' curves in Figure 4.

[2] The approximations were found to predict all trim state variables within 1% of the iterative benchmark values for forward speeds above 2.9 knots; as the vehicle speed decreased, the approximation became increasingly inaccurate, with some decision variables diverging more than 25% from the iterative solution (see Appendix C.4 for details of the comparison).

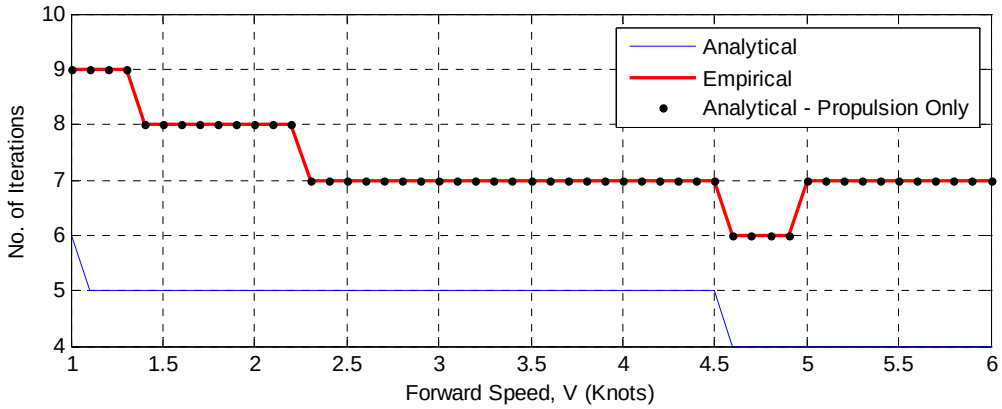

Figure 4: Comparison of Algorithm Convergence for various initial estimate cases, showing the number of iterations required to reach convergence, as defined by expressions (3.4.4) and (3.4.5).

3.5.2 Simplified Analytical Starting Point

Sgarioto's expression (3.4.8) for the starting point retains only propulsion-related terms. An equivalent expression can be derived from the analytical approximations for the trimmed state given in Appendix B. Substituting the parameters given in Appendix A into equations (C.3.1), (C.3.2) and (C.2.15), propulsion-related variables can be approximated as

$$\mathbf{x}_0 = \left[\alpha_0, \beta_0, \phi_0, \theta_0, n_0, u_{p0}, \tau_0, \delta_{s0}, \delta_{r0}\right]^T = \left[0,\, 0,\, 0,\, 0,\, 72V,\, 0.69V,\, 36V - 0.058V^2,\, 0,\, 0\right]^T. \tag{3.5.2}$$

The resulting convergence properties are shown as the 'Analytical – Propulsion Only' curves in Figure 4.

3.5.3 Results and Discussion

Figure 4 compares the convergence behaviour of the Newton-Raphson algorithm using the three methods of initialisation proposed in previous sub-sections. Initialisation using the analytical expressions given in Appendix B results in the fastest convergence at all forward speeds, but the difference between the fastest and slowest options is at most 40% or 4 iterations; therefore it is practically insignificant. However, the analytical expressions were derived from first principles and should be valid for any reasonable set of vehicle parameters; therefore they are to be preferred.

4. Parametric Study

This section describes a parametric study of the trim state of the REMUS 100 model as the forward speed V varies. The objective of this study was to assess whether the results of the iterative trim calculation are credible.

The trim algorithm developed in Section 3.4 was used to calculate the trim states of the REMUS 100 model for forward speed values from one knot up to six knots, a superset of the working range of the vehicle, which is approximately 2 knots to 5 knots. The results are plotted in Figure 5 to Figure 7 and discussed in the following sections.

4.1 Vehicle Orientation Angles

The trimmed vehicle orientation angles are plotted in Figure 5. The angles of attack and sideslip are equivalent measures of the vertical and lateral velocity components, v and w.

The angle of attack required for trim at 1 knot is about 5° nose-down and decreases non-linearly towards zero as forward speed increases. This decrease in magnitude is expected[3].

A similar trend of decreasing magnitude is visible in the sideslip angle but on a smaller scale to the trend in the angle of attack. This is because the same mechanisms are at work, but the forces exerted on the lateral axis in trimmed flight are smaller. The small amount of sideslip that is required to balance the lateral forces arises from cross-coupling effects brought on by a non-zero trimmed roll angle. The rudder and elevator fins of the REMUS vehicle are rigidly coupled and unable to provide direct compensation for the torque exerted by the rotation of the propeller, so the REMUS vehicle has a non-zero roll angle at all working speeds.

The variation of pitch angle θ with forward speed is almost identical to the variation of the angle of attack α. The difference is so small that it is not possible to differentiate them on the scale shown in the figure. This behaviour is consistent with a state of steady level flight with minimal sideslip.

The roll angle ϕ is approximately -0.2° at one knot forward speed and increases with speed to almost -5.5° at 6 knots. The increase in roll angle causes an increase in righting moment that balances the increasing propeller torque.

[3] The lifting force produced by a slim, nearly level body is approximately proportional to the product of the angle of attack and the forward speed squared. Thus, lift increases as the square of the speed of the vehicle; consequently a smaller angle of attack is required to balance the positive buoyancy of the vehicle.

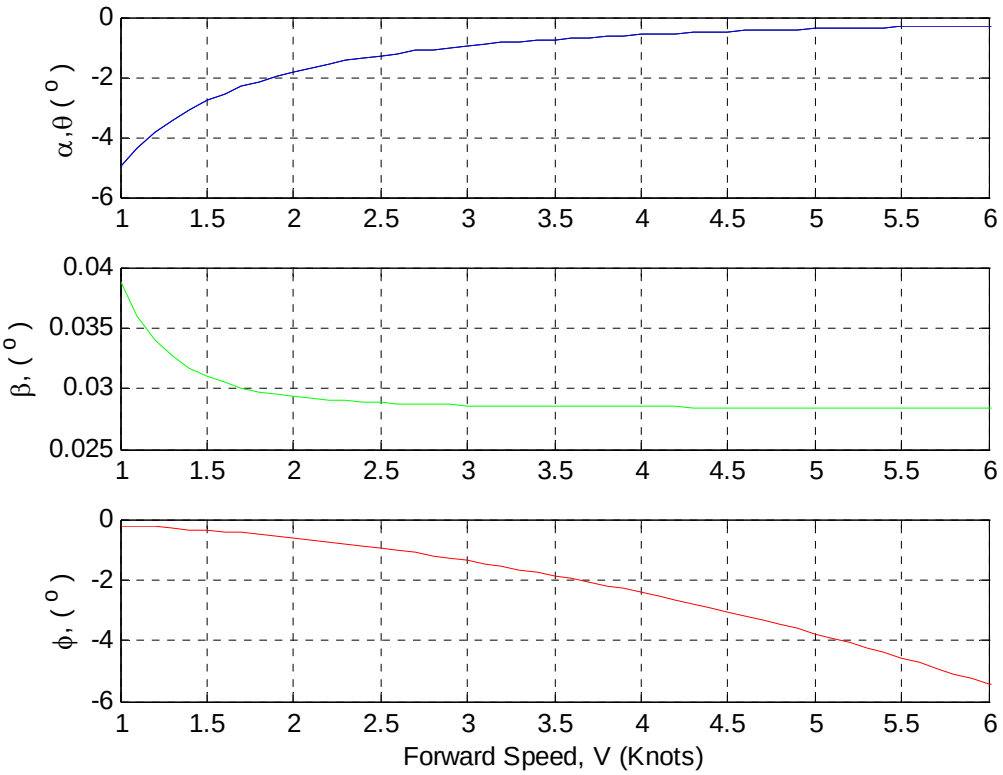

Figure 5: Variation of trimmed angle of attack/pitch angle (top), sideslip angle (middle) and roll angle (bottom) with forward speed

4.2 Propulsion Variables

As expected, the propulsion system variables - propeller rotational rate and propeller inflow speed – show an approximately linear relationship to the forward speed. In order to produce a greater forward speed, more thrust is required. This is produced by a higher propeller rotation rate, which pulls water through the propeller at a faster rate or a higher inflow speed. This is illustrated in Figure 6. The propeller speed increases at a rate of about 285 RPM per knot of increase in speed. The inflow speed increases at a rate of about 0.29 $m \cdot s^{-1}$ per knot of increase in speed.

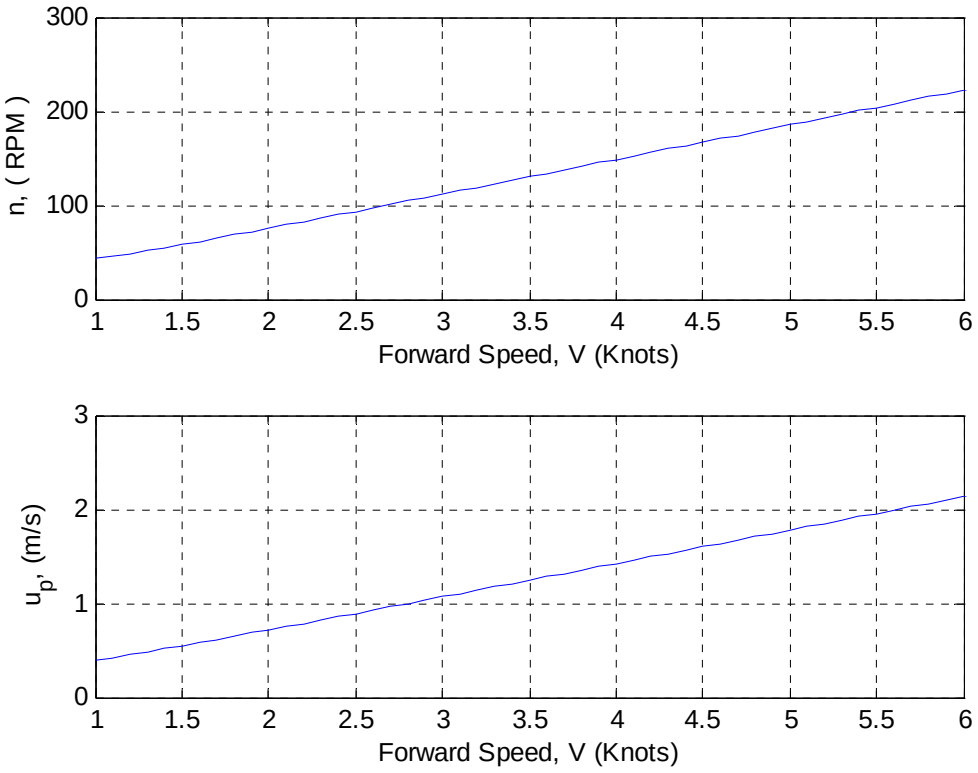

Figure 6: Variation of propulsion system state variables propeller rotation rate (top) and propeller inflow speed

4.3 Control Inputs

The behaviour of the control inputs with forward speed is illustrated in Figure 7. As with the propeller rotation rate and the propeller inflow speed, the torque generated by the propeller motor increases approximately linearly as the speed increases, rising from approximately 21 Nm at 1 knot to about 111 Nm at 6 knots.

The angle of the rudder fin δ_r has a trimmed value of -0.15° at a forward speed of 1 knot. The magnitude of this angle decreases exponentially as forward speed increases, levelling out at approximately -0.11° for speeds greater than 2.5 knots.

The control system of the DSTO REMUS 100 vehicle is typically able to maintain a state of near-equilibrium around a trimmed condition for speeds exceeding 2.5 knots. At lower speeds, the control system may lose its capacity to control depth and the vehicle begins to 'porpoise'. The modelled behaviour of the trimmed elevator fin angle is in agreement with expectation for speeds over 2 knots. However, the model predicts that the elevator fin angle reaches a minimum value of -6.5° at 1.5 knots and decreases at lower speeds, which is not intuitive; possibly downward body lift force dominates fin lift force at low speeds.

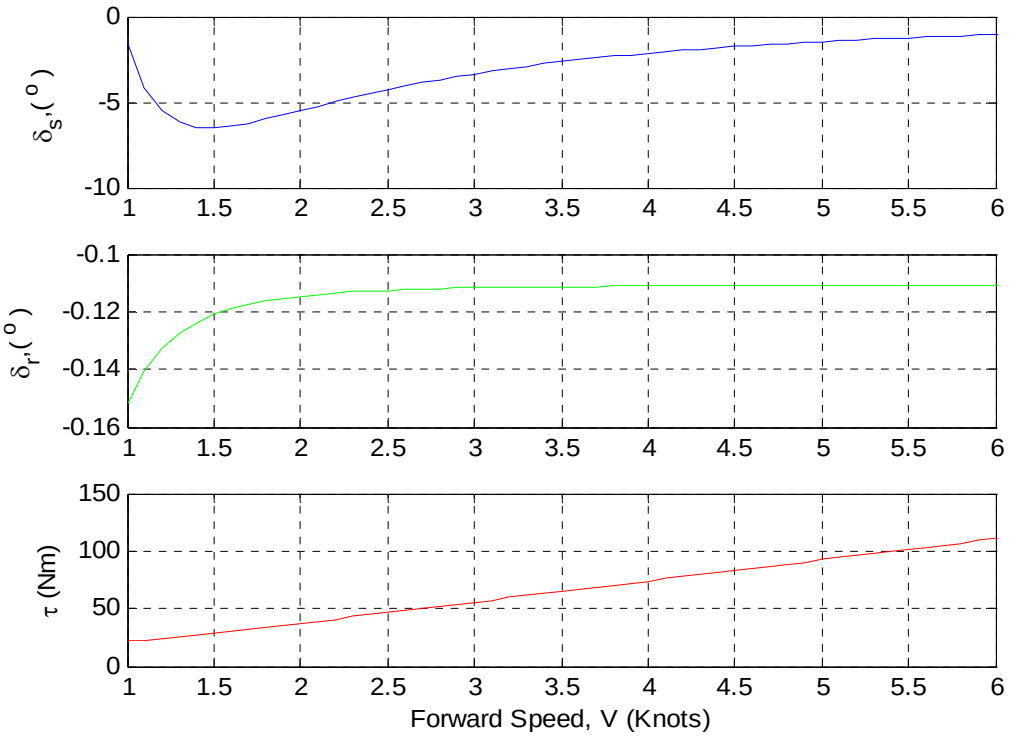

Figure 7: Variation of trimmed control input settings elevator fin angle (top), rudder fin angle (middle) and propeller motor mechanical torque (bottom) with forward speed.

In summary, the numerical trim algorithm converges over a wider range of speeds than those at which the physical vehicle operates. The predicted behaviour of the trimmed vehicle states is in agreement with expectation for speeds over two knots. The predicted behaviour of the trimmed elevator fin angle at speeds under 2 knots is not intuitive, but it is beyond the range that can be checked by reference to the behaviour of the physical vehicle.

5. Conclusion

The calculations given in this work demonstrate that the REMUS 100 vehicle dynamics model described by Prestero [1] and extended by Sgarioto [4] is readily trimmed using a numerical zero-finding procedure based on Newton-Raphson iteration. This work also presents analytical approximations to the trimmed state that can be used as a starting point for numerical procedure. With suitable parameters, the procedure should be adjustable to model different REMUS 100 configurations and vehicles of similar size and shape.

6. References

1. Prestero, T. "Verification of a Six-Degree-of-Freedom Simulation Model for the REMUS Autonomous Underwater Vehicle", MSc/ME Thesis, Massachusetts Institute of Technology, Sept. 2001

2. Sgarioto, D. "Steady State Trim and Open Loop Stability Analysis for the REMUS Autonomous Underwater Vehicle", Defence Technology Agency, New Zealand Defence Force, DTA Report 254, March 2008.

3. Press, W. H., Teukolksy, S. A., Vetterling, W. T. and Flannery, B. P. (1992) *Numerical Recipes in C*, 3rd Edition, Cambridge University Press.

4. Sgarioto, D. "Control System Design and Development for the REMUS Autonomous Underwater Vehicle", Defence Technology Agency, New Zealand Defence Force, DTA Report 250, May 2007.

Acknowledgments

This work was motivated by private communications from Dr Daniel Sgarioto, then working at the New Zealand Defence Technology Agency. The Defence Technology Agency also provided Dan's Matlab implementation of a REMUS 100 dynamical model, some of which is reproduced in this work with their permission.

Appendix A: REMUS 100 Model Parameters

A.1. Mathematical Symbols

Table 4: Symbols used to describe rigid-body dynamics in the body-fixed reference frame

Degree of Freedom	External Forces and Moments (units)	Corresponding Rates of Change (units)	Corresponding Displacements (units)
Surge	X (N)	u (m.s^{-1})	x (m)
Sway	Y (N)	v (m.s^{-1})	y (m)
Heave	Z (N)	w (m.s^{-1})	z (m)
Roll	K (N.m)	p (rad.s^{-1})	ϕ (rad)
Pitch	M (N.m)	q (rad.s^{-1})	θ (rad)
Yaw	N (N.m)	r (rad.s^{-1})	ψ (rad)

Table 5: Symbols used to describe the stability reference frame variables

Name	Symbol	Units
Vehicle speed	V	m.s^{-1}
Angle of attack	α	rad
Sideslip angle	β	rad

Table 6: Symbols used to describe the state of the propulsion system

Name	Symbol	Units
Propeller rotation rate	n	rad.s^{-1}
Propeller inflow velocity	u_p	m.s^{-1}
Control Torque	τ	N.m

Table 7: Symbols used to describe the Newton-Raphson Algorithm

Symbol	Name
$\boldsymbol{\Xi}$	Vehicle state vector
$\mathbf{u}$	Control input vector
$\mathbf{x}$	Decision variable vector
$\boldsymbol{\Omega}$	Trim variable vector
$\mathbf{J}$	Jacobian matrix
ε	Error (difference between current and previous Newton-Raphson estimate)
ε_{tol}	Error tolerance
Δx	Newton-Raphson perturbation size

A.2. Case Study Parameters

Hydroid has manufactured several different variants of the REMUS 100 AUV. The parameters used by Prestero [1] and extended by Sgarioto [4] are not applicable to the DSTO REMUS 100, which is longer, heavier and fitted with additional equipment. However, they are suitable as a basis for comparison between different approaches to calculation and they are repeated here for reference.

Table 8: General vehicle and environmental parameters

Parameter	Description	Value	Units
G	Gravitational constant	9.81	kg.m.s^{-2}
ρ	Density of sea water	1030	kg.m^{-3}
M	Mass of vehicle	30.48	kg
I_{xx}	Vehicle moment of inertia around x-axis	0.177	kg.m^2
I_{yy}	Vehicle moment of inertia around y-axis	3.45	kg.m^2
I_{zz}	Vehicle moment of inertia around z-axis	3.45	kg.m^2
z_g	z-coordinate of vehicle centre of gravity	19.6	mm
W-B	Buoyancy of vehicle	0.34	kg (0.75 lb)

Table 9: Propulsion system parameters

Parameter	Description	Value	Units
m_f	Mass of propeller control volume	0.51965	kg
K_n	Thruster motor damping coefficient	0.5	kg.m^2.s^{-1}
J_m	Thruster motor moment of inertia	1	kg.m^2
a_p	Axial flow parameter	0.25	
w_p	Thruster wake fraction number	0.2	
τ_p	Propeller thrust reduction factor	0.1	
d_{f_0}	Linear damping coefficient	6.604	
d_f	Quadratic damping coefficient	16.51	

$$d_{f_0} = \frac{-2X_{u|u|}}{(1-\tau_p)(1+a_p)(1-w_p)} = \frac{-2\times -2.972}{0.9\times 1.25\times 0.8} = 6.604.$$

$$d_f = \frac{-X_{u|u|}}{(1-\tau_p)(1+a_p)a_p(1-w_p)^2} = \frac{2.972}{0.9\times 1.25\times 0.25\times 0.64} = 16.51.$$

$$b = d_{f_0} - d_f(1-w_p)V = 16.51 - 6.604\times 0.8\times V = 16.51 - 5.283V.$$

Table 10: Non-linear drag, lift and thruster coefficients

Parameter	Description	Value	Units
$X_{u\|u\|}$	Axial drag coefficient – resisting forward motion	-2.972	kg.m^{-1}
K_{pp}	Coefficient of moment resisting roll	-0.13	kg.m.rad^{-2}
Y_{vv}	Coefficient of drag resisting sway	-1310	kg.m^{-1}
Y_{rr}	Cross-flow drag coefficient – resisting yaw	0.632	kg.m.rad^{-2}
Z_{ww}	Coefficient of drag resisting heave	-1310	kg.m^{-1}
Z_{qq}	Cross-flow drag coefficient – resisting pitch	-0.632	kg.m.rad^{-2}
M_{ww}	Coefficient of moment resisting heave	3.18	kg
M_{qq}	Coefficient of moment resisting pitch	-188	kg.m^2.rad^{-2}
N_{vv}	Coefficient of moment resisting sway	-3.18	kg
N_{rr}	Coefficient of moment resisting yaw	-94	kg.m^2.rad^{-2}
Y_{uv}	Drag resisting sway due to forward and sway motion	-28.6	kg.m^{-1}
Z_{uw}	Drag resisting heave due to forward and yaw motion	-28.6	kg.m^{-1}
M_{uw}	Coefficient of moment resisting pitch due to forward and yaw motion	24	kg
N_{uv}	Coefficient of moment resisting yaw due to forward and sway motion	-24	kg

Table 11: Thruster coefficients

Parameter	Description	Value	Units
$T_{n\|n\|}$	Thrust coefficient	6.279×10^{-4}	kg.m.rad^{-2}
$Q_{n\|n\|}$	Torque coefficient	-1.121×10^{-5}	kg.m^2.rad^{-2}

Table 12: Added mass coefficients

Parameter	Description	Value	Units
$X_{\dot{u}}$	Axial added mass	-0.93	kg
$Y_{\dot{v}}$	Crossflow added mass	-35.5	kg
$Y_{\dot{r}}$	Crossflow added mass	1.93	kg.m.rad^{-1}
$Z_{\dot{w}}$	Crossflow added mass	-35.5	kg
$Z_{\dot{q}}$	Crossflow added mass	-1.93	kg.m.rad^{-1}
$K_{\dot{p}}$	Rolling added mass	-0.0704	kg.m^2.rad^{-1}
$M_{\dot{w}}$	Crossflow added mass	-1.93	kg.m
$M_{\dot{q}}$	Crossflow added mass	-4.88	kg.m^2.rad^{-1}
$N_{\dot{v}}$	Crossflow added mass	1.93	kg.m
$N_{\dot{r}}$	Crossflow added mass	-4.88	kg.m^2.rad^{-1}

Table 13: Added mass cross term coefficients

Parameter	Description	Value	Units
X_{uq}	Drag coefficient for forward and pitching motion	-35.5	kg.rad^{-1}
X_{qq}	Drag coefficient for pitching motion	-1.93	kg.m.rad^{-2}
X_{vr}	Drag coefficient for sway and yawing motion	35.5	kg.rad^{-1}
X_{rr}	Drag coefficient for yawing motion	-1.93	kg.m.rad^{-2}
Y_{ur}	Sway coefficient for forward and yawing motion	5.22	kg.rad^{-1}
Y_{wp}	Sway coefficient for heave and rolling motion	35.5	kg.rad^{-1}
Y_{pq}	Sway coefficient for rolling and pitching motion	1.93	kg.m.rad^{-2}
Z_{uq}	Heave coefficient for forward & pitching motion	-5.22	kg.rad^{-1}
Z_{vp}	Heave coefficient for sway and rolling motion	-35.5	kg.rad^{-1}
Z_{rp}	Heave coefficient for heave & rolling motion	1.93	kg.m.rad^{-2}
M_{uq}	Pitch mom. coeff. for forward & pitching motion	-2	kg.m.rad^{-1}
M_{vp}	Pitch mom. coeff. for sway & rolling motion	-1.93	kg.m.rad^{-2}
M_{rp}	Pitch mom. coeff. for yaw & rolling motion	4.86	kg.m^2.rad^{-2}
N_{ur}	Yaw mom. coeff. for forward and yawing motion	-2	kg.m.rad^{-1}
N_{wp}	Yaw mom. coeff. for heave and rolling motion	-1.93	kg.m.rad^{-2}
N_{pq}	Yaw mom. coeff. for rolling and pitching motion	-4.86	kg.m^2.rad^{-2}

Table 14: Control surface non-linear coefficients

Parameter	Description	Value	Units
$Y_{uu\delta_r}$	Sway force coefficient for rudder displacement	9.64	kg.m^{-1}.rad^{-1}
$Z_{uu\delta_s}$	Lift coefficient for elevator displacement	-9.64	kg.m^{-1}.rad^{-1}
$M_{uu\delta_s}$	Pitch moment coefficient for elevator displacement	-6.15	kg.rad^{-1}
$N_{uu\delta_r}$	Yaw moment coefficient for rudder displacement	-6.15	kg.rad^{-1}

Appendix B: Algorithm Implementation

The trim method described in Section 3.4 was implemented as an algorithm in MATLAB code, accessible via the function *'trim.m'*. The *trim* function takes as inputs the desired forward speed V and the initial trim state estimate $\mathbf{x}_0$ and outputs the trim estimate as vector $\mathbf{x}$, the number of iterations required to reach convergence and the final error ε. The *trim* function calls a function *'eom.m'*, which implements the state-space form of the equations of motion, through an interface function '*f*'. The *eom* function and a function '*coeff*' that it invokes were written by Dr Daniel Sgarioto and are used with the permission of the New Zealand Defence Technology Agency (DTA).

B.1. Interface Function

The *trim* function is generic, but some implementation-specific adjustments were required to use the DTA routines. A sub-function '*f*' was created to interface between the input and output vectors of the *trim* function and the variables employed by the state rates function *eom*. The sub-function converts the stability axis variables (V, a, β) to body velocities (u, v, w), via the transformations (2.3.10) to (2.3.12), and translates variables between the *trim* input and output formats and the internal format used by *eom*. This process is represented in a flow diagram shown in Figure 8.

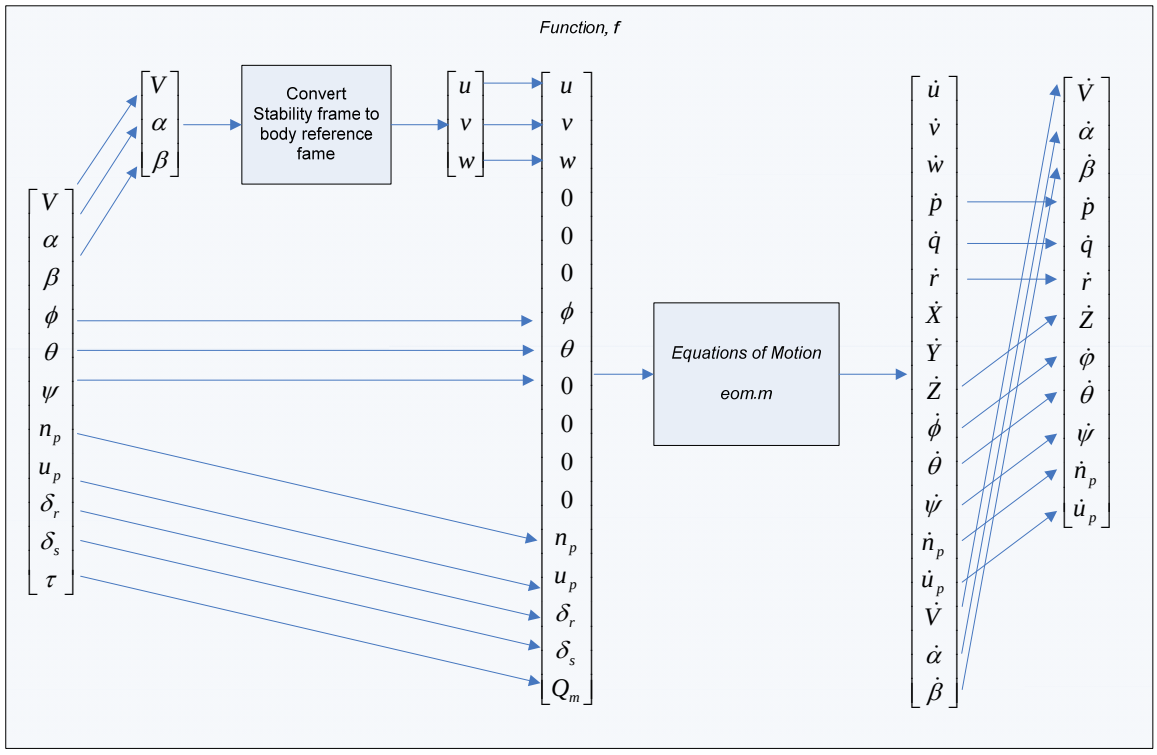

Figure 8: Flow diagram for interface function, f

B.2. Iteration Algorithm

The Newton-Raphson method is implemented in the function *'trim.m'*. The algorithm is designed to trim the equations contained within the interface function *'f'*. The algorithm is split loosely into three sections: algorithm initialisation, the Newton-Raphson iteration and post-processing and display. The Newton-Raphson iteration consists of a while-loop which continues to apply the Newtown-Raphson method while the total difference between the current and previous trim estimates is greater than a preset error tolerance value. There is also a loop break condition if the number of iterations exceeds 100; this is included to stop the algorithm from running indefinitely in the event that a trim point can not be found. A flow diagram representation of the algorithm is shown in Figure 9.

B.2.1 Jacobian matrix calculation

The Jacobian matrix is calculated within the state rates interface function *f*. The process is implemented as a for-loop which cycles through each trim variable. Within each iteration, all the partial derivatives with respect to the trim variable in question are calculated; that is, one column of the Jacobian matrix is calculated at each iteration. A flow diagram representation of the Jacobian generation loop is shown separately within Figure 9.

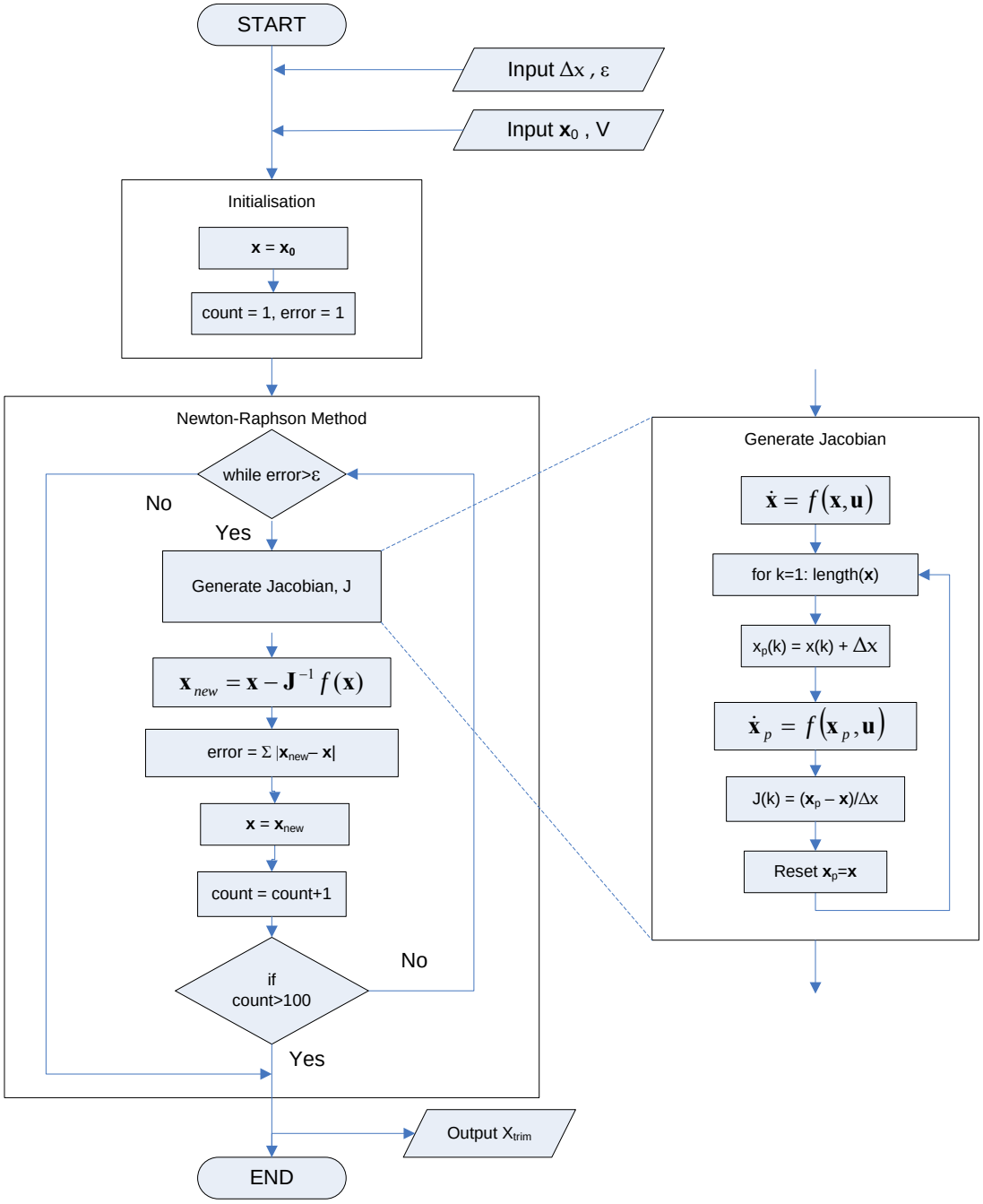

Figure 9: Trim algorithm flow diagram with the Jacobian generation process shown as an inset

B.3. Trim function MATLAB script

B.3.1 Main Trim Function *'trim.m'*

```
function   [x_trim, iterations, error] = trim(V_knots, X0)

% Setup global variables
global m Minv W B zg Ixx Iyy Izz w_prop tau Jm m_f Kn Xuu Xudot Yvdot
Nvdot Zwdot Mwdot Zqdot Mqdot Yrdot Nrdot Xwq Xqq Xvr Xrr Yvv Yrr Yuv Yur
Ywp Ypq Yuudr Zww Zqq Zuw Zuq Zvp Zrp Zuuds Kpp Mww Mqq Muw Muq Mvp Mrp
Muuds Nvv Nrr Nuv Nur Nwp Npq Nuudr Ndr Kpdot Tnn Qnn df0 df nd
global V

% Define trim conditions
V = V_knots*0.514444; % m/s

% Calculate hydro coefficients for desired speed
coeffs

%Decision variables

% Trim input vector indices
x_ind = [18 19 10 11 13 16 14 15 17];
% x = [alpha beta phi theta n_prop u_prop delta_s delta_r Qm]

% Trim output vector for generating Jacobian
xd_ind = [1 2 3 4 5 6 9 10 11 12 13 14 15 16 17];
% x_dot = [u v w p q r Z phi theta psi n_prop u_prop V alpha beta]

if nargin<2,
    % Initial trim estimate
    %X0 = [alpha0 beta0 phi0 theta0 n_prop0 up0 delta_s0 delta_r0 Qm0]
    X0 = [ 0 0 0 0 72*V 0.7*V 0 0 36*V ]; % Analytical - prop only
end

dx = 0.001; % perturbation size
x_t = X0'; % initialise trim state vector
error=100; % initialise error
tol = 1e-10; % error tolerance
counter = 1; % initialise while loop counter

%% Main Loop ------------------------------------------------------------

while error > tol

% Generate Jacobian ----------------------------------------------------
    for i = 1:length(x_t)
        x_tp = x_t;% reset trim vector
        x_tp(i) = x_t(i) + dx; %perturb i_th element in trim vector

        fx_tp = f(x_tp,x_ind,xd_ind);% Calculate perturbed x_dot
        fx_t = f(x_t,x_ind,xd_ind);% Calculate unperturbed x_dot
```

```
        J(:,i) = (fx_tp-fx_t)./dx;%calculate i_th column of Jacobian
    end
%-------------------------------------------------------------------------
    x_t_old = x_t;%save old trim state
    x_t = x_t - J\fx_t;% calculate new trim state

    error = sum(abs(x_t-x_t_old)); % calculate error between new and old
trim states
    counter = counter+1;% advance counter

    if counter>100 % Break if calculated more than 100 iterations
        break
    end
end
%%------------------------------------------------------------------------

% Final trim state
x_trim = x_t;

%calculate (u,v,w)
u_t = V.*cos(x_t(1))*cos(x_t(2));
v_t = V.*sin(x_t(2)) ;
w_t = V.*sin(x_t(1))*cos(x_t(2));
alpha_t = x_t(1);
beta_t = x_t(2);
phi_t = x_t(3);
theta_t = x_t(4);
np_t = x_t(5);
up_t = x_t(6);
Q_t = x_t(9);
ds_t = x_t(7);
dr_t = x_t(8);

save trim_state_1knots u_t v_t w_t alpha_t beta_t phi_t theta_t np_t up_t
Q_t ds_t dr_t

%% Display output
iterations = counter - 1;
fprintf('\nNo. of interations: %3.0i\n',iterations)
fprintf('final error: %1.2e\n',error)
fprintf('\nTRIM STATE\n')
fprintf('--------------\n')
fprintf('u = %6.4f m/s\n',u_t)
fprintf('v = %6.4f m/s\n',v_t)
fprintf('w = %6.4f m/s\n',w_t)
fprintf('alpha = %6.4f deg\n',x_t(1)*180/pi)
fprintf('beta = %6.4f deg\n',x_t(2)*180/pi)
fprintf('phi = %6.4f deg\n',x_t(3)*180/pi)
fprintf('theta = %6.4f deg\n',x_t(4)*180/pi)
fprintf('n_prop = %6.4f RPM\n',x_t(5)*60/2/pi)
fprintf('u_prop = %6.4f m/s\n',x_t(6))
fprintf('Q = %6.4f Nm\n',x_t(9))
fprintf('delta_s = %6.4f deg\n',x_t(7)*180/pi)
fprintf('delta_r = %6.4f deg\n',x_t(8)*180/pi)
fprintf('--------------\n')
```

```
%%-------------------------------------------------------------------------

function xdot = f(x,x_ind,xd_ind)
% Function to manipulate the variables passed to and from the equation of
% motion function - eom.m

X = zeros(19,1); % initialise vehicle state vector
X([x_ind]) = x; % insert trim states into appropriate elements of vehicle
state vector

Xdot = eom(X); % calculate vehicle state rate vector

xdot = Xdot([xd_ind]); % extract appropriate elements vehicle state rates
vector for Jacobian

%%-------------------------------------------------------------------------
```

B.3.2 Equations of Motion Function *'eom.m'*

```
 function  [xdot] = eom(x)

% Extracts of code taken from
% Daniel Sgarioto, New Zealand Defence Technology Agency
% Nov 2006

global m Minv W B zg Ixx Iyy Izz w_prop tau Jm m_f Kn Xuu Xudot Yvdot
Nvdot Zwdot Mwdot Zqdot Mqdot Yrdot Nrdot Xwq Xqq Xvr Xrr Yvv Yrr Yuv Yur
Ywp Ypq Yuudr Zww Zqq Zuw Zuq Zvp Zrp Zuuds Kpp Mww Mqq Muw Muq Mvp Mrp
Muuds Nvv Nrr Nuv Nur Nwp Npq Nuudr Ndr Kpdot Tnn Qnn df0 df nd

global V

% Mass Matrix
M(1,1) = m - Xudot;
M(1,5) = m*zg;
M(1,6) = 0;
%
M(2,2) = m - Yvdot;
M(2,4) = -m*zg;
M(2,6) =  - Yrdot;
%
M(3,3) = m - Zwdot;
M(3,4) = 0;
M(3,5) =  - Zqdot;
%
M(4,2) = -m*zg;
M(4,3) = 0;
M(4,4) = Ixx - Kpdot;
%
M(5,1) = m*zg;
M(5,3) =  - Mwdot;
M(5,5) = Iyy - Mqdot;
%
M(6,1) = 0;
```

```
M(6,2) =  - Nvdot;
M(6,6) = Izz - Nrdot;
%
Minv = inv(M);

% Get state variables
alpha = x(18);
beta = x(19);

% Forward speed constraint----
u = V.*cos(alpha)*cos(beta);
v = V.*sin(beta) ;
w = V.*sin(alpha)*cos(beta);
%------------------------------
p = x(4);
q = x(5);
r = x(6);

xpos = x(7);
ypos = x(8);
zpos = x(9);

phi = x(10);
theta = x(11);
psi = x(12);

nprop = x(13);
delta_s = x(14);
delta_r = x(15);
up = x(16);
Qm = x(17);

ua = (1 - w_prop)*u;
Uc = [0 0];
%%
% Hydrodynamic Thrust & Torque
%
T = Tnn*abs(nprop)*nprop;
Q = Qnn*abs(nprop)*nprop;
%Qm = x(17); defined above

% Begin EOM
%
c1 = cos(phi); c2 = cos(theta); c3 = cos(psi); s1 = sin(phi); s2 =
sin(theta); s3 = sin(psi); t2 = tan(theta);
%
% Set total forces from equations of motion
%
Xf = -(W-B)*sin(theta) + Xuu*u*abs(u) + (Xwq-m)*w*q + (Xqq)*q^2 ...
    + (Xvr+m)*v*r + (Xrr)*r^2 - m*zg*p*r + (1 - tau)*T;
%
Yf = (W-B)*cos(theta)*sin(phi) + Yvv*v*abs(v) + Yrr*r*abs(r) + Yuv*u*v...
    + (Ywp+m)*w*p + (Yur-m)*u*r - (m*zg)*q*r + (Ypq)*p*q ...
    + Yuudr*u^2*delta_r ;
%
```

```
Zf = (W-B)*cos(theta)*cos(phi) + Zww*w*abs(w) + Zqq*q*abs(q)+ Zuw*u*w ...
    + (Zuq+m)*u*q + (Zvp-m)*v*p + (m*zg)*p^2 + (m*zg)*q^2 ...
    + (Zrp)*r*p + Zuuds*u^2*delta_s ;
%
Kf =  - (zg*W)*cos(theta)*sin(phi) ...
    + Kpp*p*abs(p) - (Izz-Iyy)*q*r - (m*zg)*w*p + (m*zg)*u*r + Q;
%
Mf = -(zg*W)*sin(theta) + Mww*w*abs(w) ...
    + Mqq*q*abs(q) + (Mrp - (Ixx-Izz))*r*p + (m*zg)*v*r - (m*zg)*w*q ...
    + (Muq)*u*q + Muw*u*w + (Mvp)*v*p ...
    + Muuds*u^2*delta_s ;
%
Nf = Nvv*v*abs(v) + Nrr*r*abs(r) + Nuv*u*v ...
    + (Npq - (Iyy-Ixx))*p*q + (Nwp)*w*p + (Nur)*u*r ...
    + Nuudr*u^2*delta_r ;
%
FORCES = [Xf Yf Zf Kf Mf Nf]' ;
%
xdot = ...

[Minv(1,1)*Xf+Minv(1,2)*Yf+Minv(1,3)*Zf+Minv(1,4)*Kf+Minv(1,5)*Mf+Minv(1,
6)*Nf

Minv(2,1)*Xf+Minv(2,2)*Yf+Minv(2,3)*Zf+Minv(2,4)*Kf+Minv(2,5)*Mf+Minv(2,6
)*Nf

Minv(3,1)*Xf+Minv(3,2)*Yf+Minv(3,3)*Zf+Minv(3,4)*Kf+Minv(3,5)*Mf+Minv(3,6
)*Nf

Minv(4,1)*Xf+Minv(4,2)*Yf+Minv(4,3)*Zf+Minv(4,4)*Kf+Minv(4,5)*Mf+Minv(4,6
)*Nf

Minv(5,1)*Xf+Minv(5,2)*Yf+Minv(5,3)*Zf+Minv(5,4)*Kf+Minv(5,5)*Mf+Minv(5,6
)*Nf

Minv(6,1)*Xf+Minv(6,2)*Yf+Minv(6,3)*Zf+Minv(6,4)*Kf+Minv(6,5)*Mf+Minv(6,6
)*Nf
    c3*c2*u + (c3*s2*s1-s3*c1)*v + (s3*s1+c3*c1*s2)*w + Uc(1)
    s3*c2*u + (c1*c3+s1*s2*s3)*v + (c1*s2*s3-c3*s1)*w + Uc(2)
    -s2*u + c2*s1*v + c1*c2*w
    p + s1*t2*q + c1*t2*r
    c1*q - s1*r
    s1/c2*q + c1/c2*r] ;
%
xdot(13,1) = (1/Jm)*(Qm - Q - (Kn*nprop));
%
xdot(14,1) = (1/m_f)*(T - df0*up - df*abs(up)*(up - ua));
%
ydotV = xdot(1,1)*cos(alpha)*cos(beta) + xdot(2,1)*sin(beta) +
xdot(3,1)*sin(alpha)*cos(beta);
%
ydotA = (xdot(3,1)*cos(alpha) - xdot(1,1)*sin(alpha))/(V*cos(beta));
%
ydotB = (1/V)*(-xdot(1,1)*cos(alpha)*sin(beta) + xdot(2,1)*cos(beta) -
xdot(3,1)*sin(alpha)*sin(beta));
%
xdot(15:17,1) = [ydotV ydotA ydotB];
```

B.3.3 Hydrodynamics Coefficients script file '*coeffs.m*'

```
% REMUS Hydrodynamic Coefficients
% Daniel Sgarioto, DTA
% Nov 2006
global V scale
% Vehicle Parameters
%
U0=V;
m = 30.48;
g = 9.81;
%
W = m*g;
B = W + (0.75*4.44822162);
L = 1.3327;
%
zg = 0.0196;
%
Ixx  = 0.177;
Iyy  = 3.45;
Izz  = 3.45;
%
cdu = 0.2;
rho = 1030;
Af = 0.0285;
d = 0.191;
xcp = 0.321;
Cydb = 1.2;
%
mq = 0.3;
%
cL_alpha = 3.12;
Sfin = 0.00665;
xfin = -0.6827;
%
gamma = 1;
a_prop = 0.25;
w_prop = 0.2;
tau = 0.1;
Jm = 1;
%
l_prop = 0.8*0.0254;
d_prop = 5.5*0.0254;
A_prop = (pi/4)*(d_prop^2);
m_f = gamma*rho*A_prop*l_prop;
%
Kn = 0.5;
%
% Most are Prestero's estimates, but revised values of some linear
% coefficients are due to Fodrea. Thruster coeffs based on results
reported by Allen et al.
%
Xwq= -35.5;
Xqq= -1.93;
Xvr= 35.5;
```

```
Xrr= -1.93;
Yvv= -1310.0;
Yrr= 0.632;
Yuv= -28.6;
Yur= 5.22;
Ywp= 35.5;
Ypq= 1.93;
Yuudr= 9.64;
Zww= -1310.0;
Zqq= -0.632;
Zuw= -28.6;
Zuq= -5.22;
Zvp= -35.5;
Zrp= 1.93;
Zuuds= -9.64;
Kpp= -0.130;
Mww= 3.18;
Mqq= -188;
Muw= 24.0;
Muq= -2.0;
Mvp= -1.93;
Mrp= 4.86;
Muuds= -6.15;
Nvv= -3.18;
Nrr= -94.0;
Nuv= -24.0;
Nur= -2.0;
Nwp= -1.93;
Npq= -4.86;
Nuudr= -6.15*scale;
Kpdot= -0.0704;
%
Xuu = -0.5*rho*cdu*Af;
Xu = -rho*cdu*Af*U0;
%
% Added Mass Coeffs
Xudot= -0.93;
%
Yvdot= -35.5;
Yrdot= 1.93;
%
Zwdot= -35.5;
Zqdot= -1.93;
%
Mwdot= -1.93;
Mqdot= -4.88;
%
Nvdot= 1.93;
Nrdot= -4.88;
%
% Added Mass Terms
%
Zwc = -15.7;
Zqc = 0.12;
%
Mwc = -0.403;
Mqc = -2.16;
```

```
%
% Added Mass Coupling Terms
Xqa = Zqdot*mq;
Zqa = -Xudot*U0;
Mwa = -(Zwdot - Xudot)*U0;
Mqa = -Zqdot*U0;
%
% Body Lift Contribution
%
Zwl = -0.5*rho*(d^2)*Cydb*U0;
%
Mwl = -0.5*rho*(d^2)*Cydb*xcp*U0;
%
% Fin Contribution
%
Zwf = -0.5*rho*cL_alpha*Sfin*U0;
Zqf = 0.5*rho*cL_alpha*Sfin*xfin*U0;
%
Mwf = 0.5*rho*cL_alpha*Sfin*xfin*U0;
Mqf = -0.5*rho*cL_alpha*Sfin*(xfin^2)*U0;
%
% Dive Plane Coeffs
Zw = Zwc + Zwl + Zwf;
Zq = 2.2;
%
Mw = -9.3;
Mq = Mqc +Mqa +Mqf;
%
% Steering Coeffs
Yv = Zw;
Yr = 2.2;
%
Nv = -4.47;
Nr = Mq;
%
% Control Surface Coeffs
Zds = -rho*cL_alpha*Sfin*(U0^2);
Mds = rho*cL_alpha*Sfin*xfin*(U0^2);
%
Ydr = -Zds/3.5;
Ndr = Mds/3.5;
%
% Thruster Coeffs
Tnn = 6.279e-004;
Tnu = 0;
%
Qnn = -1.121e-005;
Qnu =  0;
%
Tnu0 = (1/(a_prop + 1))*Tnu;
Qnu0 = (1/(a_prop + 1))*Qnu;
%
df0 = (-Xu)/((1 - tau)*(1 + a_prop)*(1 - w_prop));
df =  (-Xuu)/((1 - tau)*(1 + a_prop)*a_prop*((1 - w_prop)^2));
%
```

Appendix C: Analytical Trim Estimation using Force Balance Conditions

Sgarioto [2] derived the initial conditions (3.4.8) for numerical solution of the trim equations empirically. It is possible to derive more accurate initial conditions from first principles. Consider the equations of motion described in Section 2.4. In the trimmed condition, we require that $p = q = r = 0$, $\dot{p} = \dot{q} = \dot{r} = 0$ and $\dot{u} = \dot{v} = \dot{w} = 0$. Consequently, the sums of forces and moments are zero, that is,

$$\Sigma X = \Sigma Y = \Sigma Z = 0 \text{ and } \Sigma K = \Sigma M = \Sigma N = 0.$$

When the vehicle is in a trimmed state, the force and moment equations simplify, giving

$$\Sigma X = (B-W)\sin\theta + X_{u|u|}u|u| + (1-\tau_p)T_{n|n|}n|n| = 0 \quad \text{(C.1)}$$

$$\Sigma Y = (W-B)\cos\theta\sin\phi + Y_{uv}uv + Y_{v|v|}v|v| + Y_{uu\delta_r}u^2\delta_r = 0 \quad \text{(C.2)}$$

$$\Sigma Z = (W-B)\cos\theta\cos\phi + Z_{w|w|}w|w| + Z_{uw}uw + Z_{uu\delta_s}u^2\delta_s = 0 \quad \text{(C.3)}$$

and

$$\Sigma K = -z_g W\cos\theta\sin\phi + Q_{n|n|}n|n| = 0 \quad \text{(C.4)}$$

$$\Sigma M = -z_g W\sin\theta + M_{uw}uw + M_{w|w|}w|w| + M_{uu\delta_s}u^2\delta_s = 0 \quad \text{(C.5)}$$

$$\Sigma N = N_{uv}uv + N_{v|v|}v|v| + N_{uu\delta_r}u^2\delta_r = 0. \quad \text{(C.6)}$$

The equations of motion (2.4.7), (2.4.8) for the thruster also simplify, giving

$$\tau = K_n n + Q_{n|n|}n|n| \quad \text{(C.7)}$$

$$d_{f_0}u_p + d_f|u_p|\left[u_p - (1-w_p)u\right] = T_{n|n|}n^2 \quad \text{(C.8)}$$

where $d_{f_0} = \dfrac{-2X_{u|u|}V}{(1-\tau_p)(1+a_p)(1-w_p)}$ and $d_f = \dfrac{-X_{u|u|}}{(1-\tau_p)(1+a_p)(1-w_p)}$.

C.1. Senses of Motion

For convenience, we adopt a number of sign conventions to simplify the absolute value quantities appearing in Equations (C.1) to (C.8). In the trimmed condition, we assume that:

- The vehicle is moving forward; that is, $|u| = u$.
- The axial water inflow velocity to the propeller is positive; that is, $|u_p| = u_p$.

- The propeller is rotating clockwise (positive); that is, $|n| = n$.
- The vehicle is passively stable in roll and the roll moment counteracts the torque due to the motion of the propeller; thus, it is negative: $\phi < 0$.
- The vehicle is positively buoyant and the buoyancy of the vehicle is counteracted by a downward lift force, which is generated by swimming the vehicle nose-down; that is, $\theta < 0$.
- Because the vehicle is slightly nose-down and moving forward, the body-frame vertical velocity is negative (upward); that is, $w < 0$ and $|w| = -w$.
- From (C.6) and the coefficient values in Appendix A, we can show that the sign of v is opposite to the sign of δ_r. Using this and the signs of θ and ϕ we previously identified, we can show from (C.2) that $v > 0$.
- It follows that $\delta_r < 0$.
- From (C.5) and the coefficients in Appendix A and noting that $w > 0$, we can show that $\delta_s > 0$.

C.2. Approximation to Nearly Level Motion

Equations (C.1) to (C.8) are non-linear and do not have an obvious general solution. However, if we assume that a vehicle in the trimmed condition will be nearly horizontal and will have its body axis oriented nearly parallel to the direction of motion, then approximations can be made that allow the equations to be solved.

The relationship between body-frame velocity and Earth-frame velocity or ground speed is given by the inverse of transformation (2.3.1) to (2.3.3), which is

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \sin\phi\cos\theta \\ \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi & \cos\phi\cos\theta \end{bmatrix} \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix}. \tag{C.2.1}$$

In the trim state, the rate of climb in the Earth frame is zero; hence, the true speed V in the stability frame is also the forward speed. Without loss of generality, we can define

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix} \equiv \begin{bmatrix} V \\ 0 \\ 0 \end{bmatrix}, \tag{C.2.2}$$

in which case Equations (C.2.1) become

$$u = V\cos\theta\cos\psi \tag{C.2.3}$$

$$v = V(\sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi) \quad \text{(C.2.4)}$$
$$w = V(\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi). \quad \text{(C.2.5)}$$

We assume that roll, pitch and (without loss of generality) yaw angles will all be small; that is, in radians,

$$\phi, \theta, \psi << 1.$$

Then to lowest order in angular quantities, we have

$$u = V(1 + o(\theta^2) + o(\psi^2) \sim V \quad \text{(C.2.6)}$$
$$v = V(\psi + o(\phi\theta)) \sim V\psi \quad \text{(C.2.7)}$$
$$w = V(\theta + o(\phi\psi)) \sim V\theta. \quad \text{(C.2.8)}$$

Note also that $v, w << u$.

Adopting the sense conventions from Section C.1 and dropping terms beyond second order in v, w, θ and ϕ, equations (C.1) to (C.3) become

$$0 = (B - W)\theta + X_{u|u|}V^2 + (1 - \tau_p)T_{n|n|}n^2 \quad \text{(C.2.9)}$$
$$0 = (W - B)\phi + Y_{uv}Vv + Y_{v|v|}v^2 + Y_{uu\delta_r}V^2\delta_r \quad \text{(C.2.10)}$$
$$0 = (W - B)[1 - 0.5\theta^2 - 0.5\phi^2] - Z_{w|w|}V^2\theta^2 + Z_{uw}V^2\theta + Z_{uu\delta_s}V^2\delta_s. \quad \text{(C.2.11)}$$

Equations (C.4) to (C.6) become

$$0 = -z_g W\phi + Q_{n|n|}n^2 \quad \text{(C.2.12)}$$
$$0 = (-z_g W + M_{uw}V^2)\theta - M_{w|w|}V^2\theta^2 + M_{uu\delta_s}V^2\delta_s \quad \text{(C.2.13)}$$
$$0 = VN_{uv}v + V^2N_{uu\delta_r}\delta_r + N_{v|v|}v^2. \quad \text{(C.2.14)}$$

Equations (C.7) and (C.8) become quadratic equations, that is,

$$\tau = K_n n + Q_{n|n|}n^2 \quad \text{(C.2.15)}$$
$$d_f u_p^2 + (d_{f_0} - d_f(1 - w_p)V)u_p - T_{n|n|}n^2 = 0. \quad \text{(C.2.16)}$$

C.3. Derivation of Analytical Approximations

C.3.1 Propulsion Subsystem

We previously approximated the forward body velocity by the forward speed, that is, $u \approx V$. In order to proceed, we need to be able to relate forward speed to propeller rotation rate. In (C.2.9), we note that the term related to buoyancy is much smaller than the term due to hydrodynamic drag at normal operating velocities, since we expect that

$$B - W = o(1), \quad X_{u|u|} = o(1), \quad V^2 = o(1) \text{ but } \theta << 1.$$

Dropping the buoyancy term, the approximate propeller revolution rate is determined by

$$\bar{n}^2 \approx -\frac{X_{u|u|} V^2}{(1-\tau_p) T_{n|n|}}. \tag{C.3.1}$$

In (C.3.1) and following equations, an overbar indicates an estimated quantity.

The estimated control torque $\bar{\tau}$ follows directly from Equation (C.2.15). The axial flow velocity also follows as the positive root of Equation (C.2.16), that is,

$$\bar{u}_p = \frac{1}{2d_f}\left[-b + \sqrt{b^2 + 4d_f T_{n|n|} \bar{n}^2}\right] \text{ where } b \equiv d_{f_0} - d_f (1 - w_p) V. \tag{C.3.2}$$

C.3.2 Manoeuvre Subsystem

Estimation of the propeller revolution rate enables estimation of roll, pitch and control surface angles. From (C.2.12), the estimated roll angle is

$$\bar{\phi} = \frac{Q_{n|n|}}{W z_g} \bar{n}^2. \tag{C.3.3}$$

Substitution of estimated roll as a known parameter in (C.2.10) enables expression of the rudder angle in terms of the sideslip velocity, as

$$\delta_r = \frac{-1}{Y_{uu\delta_r} V^2}\left[(W - B)\bar{\phi} + Y_{uv} V v + Y_{v|v|} v^2\right] \tag{C.3.4}$$

Substitution of (C.3.4) in (C.2.14) results in a quadratic expression as following; the sway velocity estimate will be the positive root.

$$0 = (B - W)\frac{N_{uu\delta_r}}{Y_{uu\delta_r}}\bar{\phi} + \left[N_{uv} - \frac{N_{uu\delta_r} Y_{uv}}{Y_{uu\delta_r}}\right] V \bar{v} + \left[N_{v|v|} - \frac{N_{uu\delta_r} Y_{v|v|}}{Y_{uu\delta_r}}\right] \bar{v}^2. \tag{C.3.5}$$

The rudder angle follows from (C.2.14), as

$$\bar{\delta}_r = \frac{-1}{N_{uu\delta_r} V^2}\left[N_{v|v|}\bar{v}^2 + N_{uv} V \bar{v}\right] \tag{C.3.6}$$

Similarly, substitution of the estimated roll in (C.2.11) allows expression of the elevator angle in terms of pitch, as

$$\bar{\delta}_s = \frac{-1}{Z_{uu\delta_s} V^2}\left[(W-B)\left[1-0.5\bar{\phi}^2\right] + Z_{uw} V^2 \theta + \left[0.5(B-W) - Z_{w|w|} V^2\right]\theta^2\right] \tag{C.3.7}$$

Substituting (C.3.7) in (C.2.13) yields a quadratic expression for the pitch estimate, as

$$\begin{aligned} 0 = {} & \frac{M_{uu\delta_s}}{Z_{uu\delta_s}}(W-B)\left[1-0.5\bar{\phi}^2\right] + \left(\frac{Z_{uw} M_{uu\delta_s}}{Z_{uu\delta_s}} V^2 + W z_g - M_{uw} V^2\right)\bar{\theta} \\ & + \left(M_{w|w|} V^2 - \frac{M_{uu\delta_s} Z_{w|w|}}{Z_{uu\delta_s}} V^2 + 0.5\frac{M_{uu\delta_s}}{Z_{uu\delta_s}}(B-W)\right)\bar{\theta}^2. \end{aligned} \tag{C.3.8}$$

The pitch estimate is the negative root of the quadratic. The estimated elevator angle follows directly from (C.3.7).

Together, the equations in this section approximate the state of the vehicle in a trimmed condition.

C.4. Comparison of Numerical and Analytical Trim Estimates

Physical intuition suggests that the accuracy of equations (C.3.1) to (C.3.8) should improve with increasing forward speed. In order to compare the present approach to estimating the trim conditions with the iterative approach adopted in Section 3.4, we substitute the coefficient values from Appendix A into the equations and evaluate them.

In order to compare with the case-study of Sgarioto [2], we set V = 4 knots and take the buoyancy force as 0.75 pounds weight, that is, 0.75 × 0.45 × 9.81 = 3.3 N. The results are shown in Table 15. By inspection, the state variables estimated with the analytical approximation are mostly in agreement with the 'benchmark' values estimated by Sgarioto or using the Newton-Raphson method described in Section 3.4, to two or three significant figures.

Table 15: Comparison of numerical (SNOPT) and analytical approximations to the trim state

Variable	Symbol	Numerical Benchmark	Analytical Approximation
Body-frame surge velocity	u	2.0575 ms^{-1}	2.0577 ms^{-1}
Body-frame sway velocity	v	0.001 ms^{-1}	0.0010 ms^{-1}
Body-frame heave velocity	w	-0.021 ms^{-1}	-0.0209 ms^{-1}
Earth-frame roll	ϕ	-2.42°	-2.4107°
Earth-frame pitch	θ	-0.583°	-0.5825°
Propeller rotation rate	N	1418 rpm	1416 rpm
Propeller inflow rate	u_p	1.42 ms^{-1}	1.4197 ms^{-1}
Angle of attack	α	-0.58°	-0.5825°
Angle of sideslip	β	0.028°	0.0284°
Control Torque	τ	74 Nm	73.9089 Nm
Elevator angle	δ_s	-2.15°	-2.1452°
Rudder angle	δ_r	-0.11°	-0.1108°

For a broader comparison, analytical trim approximations were calculated for forward speeds from 1 to 6 knots and compared against values obtained from the iterative Newton-Raphson algorithm developed in Section 3.4. The results are illustrated in Figure 10 to Figure 12. The relative percentage differences between the results of the two solution methods are plotted in Figure 13 to Figure 15 for further clarification.

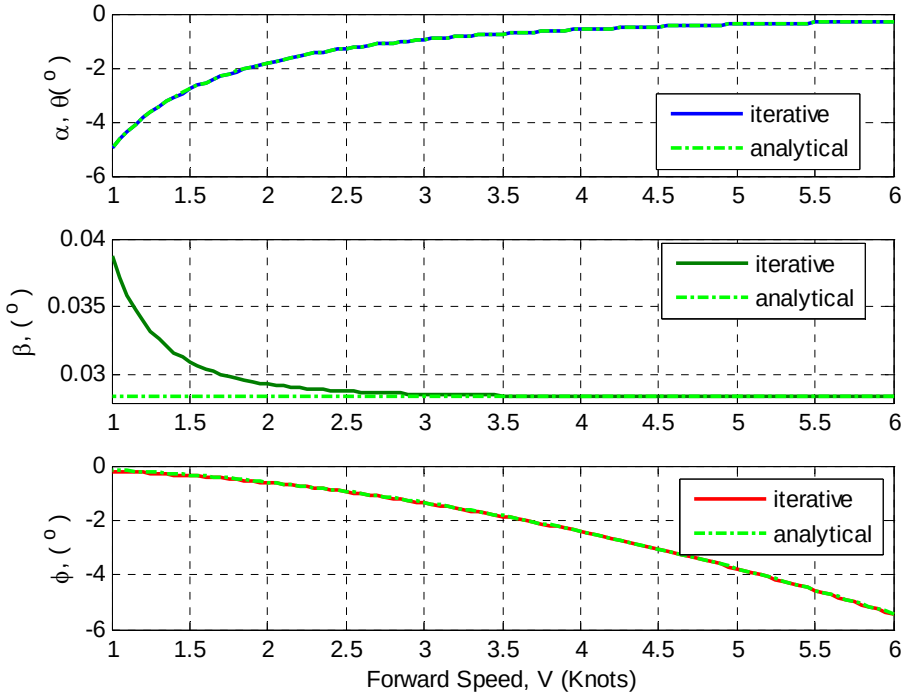

Figure 10: Comparison of the iterative solution and analytical approximation for the trim-state angles of attack or pitch (top), sideslip (middle) and roll (bottom)

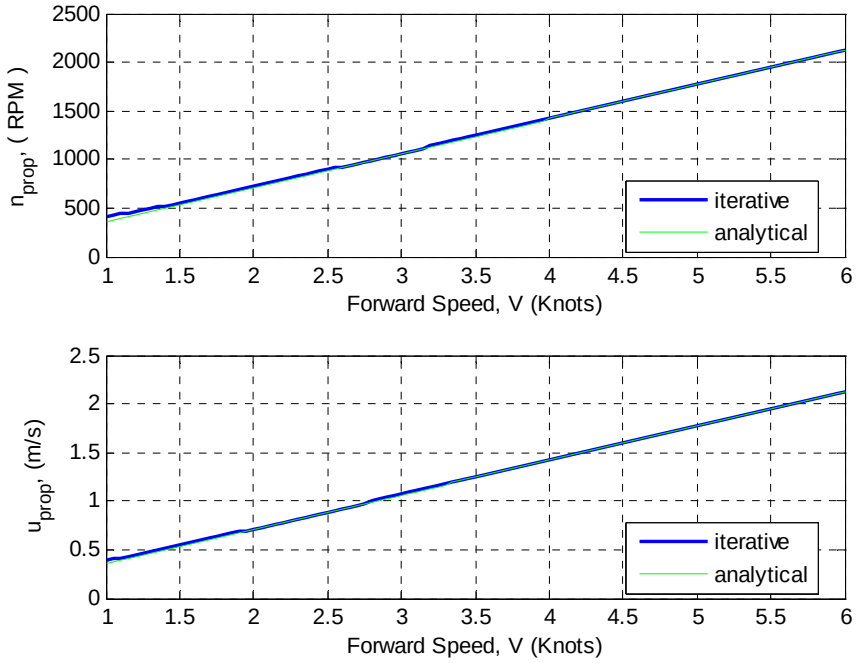

Figure 11: Comparison of iterative and analytical solutions for the propulsion system variables - propeller rotation rate (top) and propeller inflow speed (bottom) versus forward speed

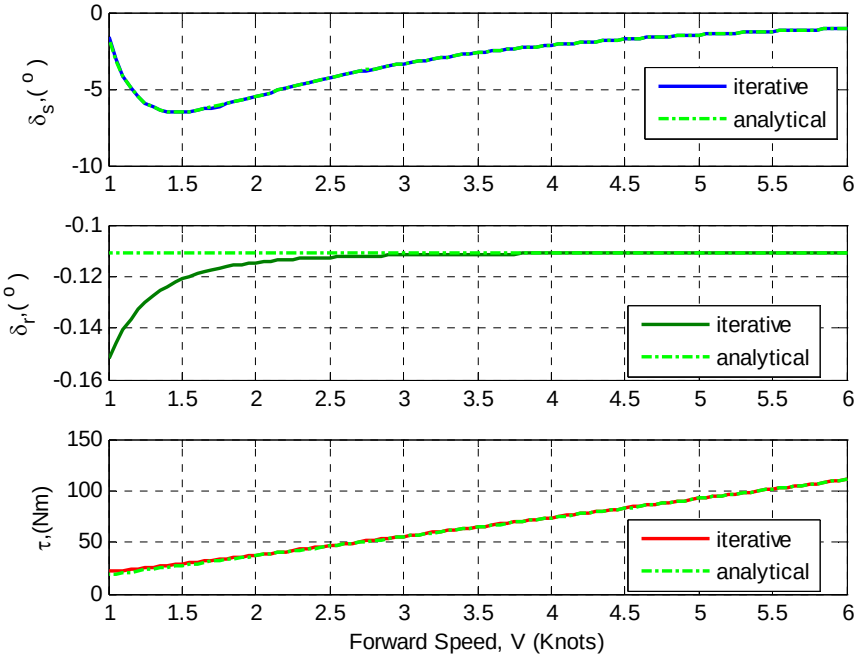

Figure 12: Comparison of the iterative and analytical solution for the control inputs - elevator angle (top), rudder angle (middle) and torque (bottom) versus forward speed

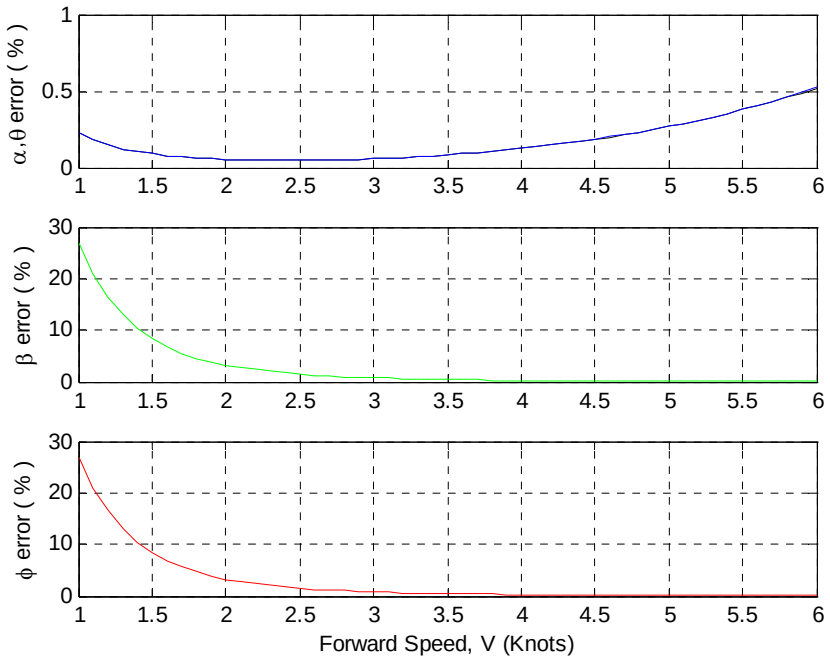

Figure 13: Relative differences between iterative and analytical solutions for the trim-state angle of attack/pitch angle (top), sideslip angle (middle) and roll angle (bottom) versus forward speed

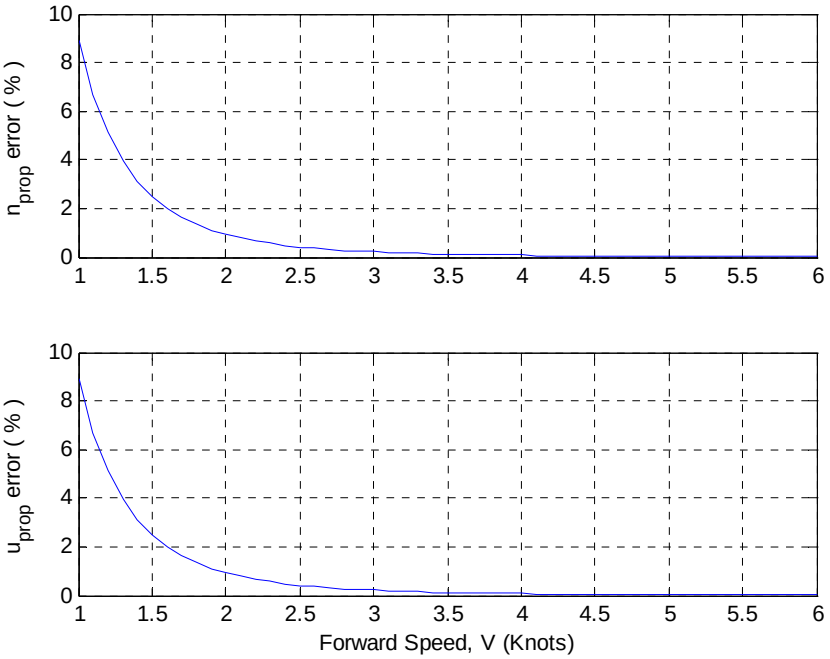

Figure 14: Relative differences between iterative and analytical solutions for the trim-state propulsion system variables - propeller rotation rate (top) and propeller inflow speed (bottom) versus forward speed

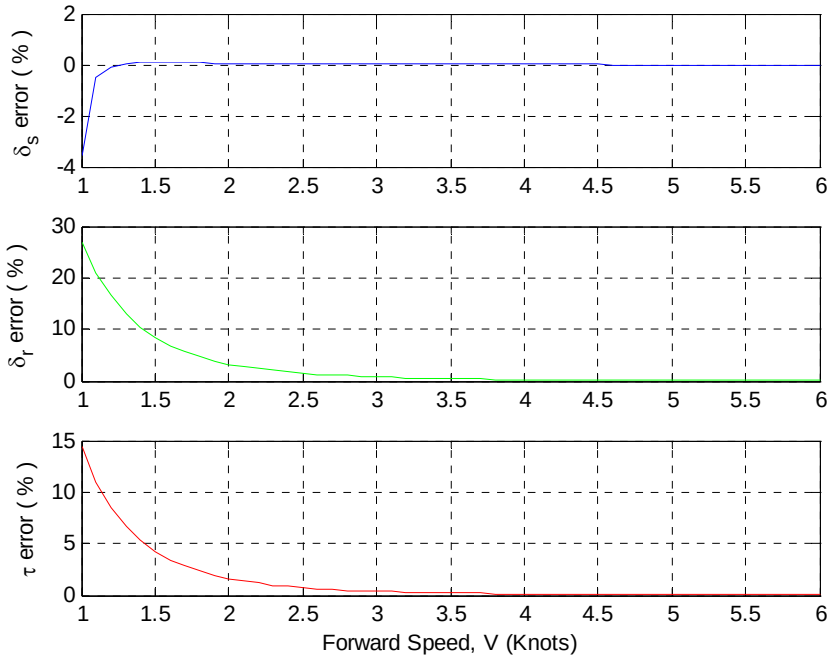

Figure 15: Relative differences between the iterative solution and analytical approximation for the trim-state control inputs - elevator fin angle (top), rudder fin angle (middle) and mechanical motor torque (bottom) versus forward speed

Appendix D: Convergence Sensitivity Analysis

In order to test the robustness of the numerical trim-estimation algorithm, a sensitivity analysis of the convergence behaviour was conducted with respect to the values of the hydrodynamic coefficients.

The robustness of the algorithm to different vehicle models was tested by altering the values of five primary hydrodynamic coefficients: M_{uw}, $M_{uu\delta_s}$, $N_{uu\delta_r}$, $X_{u|u|}$ and Z_{uw}. This set consists of coefficients which Sgarioto previously identified as having a significant impact on the longitudinal (x-z plane) stability of the REMUS model [2]. Each of the parameters is also sensitive to the physical dimensions of the vehicle.

A parametric analysis was performed by varying each of the five coefficients individually. This was done by running the trim algorithm for a forward speed of 4 knots with the value of the coefficient at 50%, 75%, 100%, 125% and 150% of the original value provided in A.1. The performance of the algorithm was compared using the number of iterations it took in each case to reach the preset error tolerance. It should be noted that, while speed of convergence has been used here as a measure of robustness, a comprehensive analysis would employ methods that consider the complete topology of the characteristic surface.

It was found that all the cases resulted in convergence to a trim solution and that in most cases the same number of iterations was required to converge as for the benchmark case. In a few cases, convergence occurred after one less iteration than the benchmark.

The trim variable results are summarised for each coefficient in Table 16 to Table 20 along with the corresponding number of iterations taken to reach convergence. As the values of the many trim variables did not change significantly compared to the original numerical benchmark solution, the trim variable results are listed as a percentage difference from the original solution. For example, the percentage difference in the angle of attack when the coefficient is reduced to 50% of its original value is calculated as follows:

$$percent\_difference = \frac{\alpha_{100\%} - \alpha_{50\%}}{\alpha_{100\%}} \times 100$$

As expected, the results show that only the variables associated with the mode of motion affected by the coefficient in question will change. For example, Table 16 shows that $M_{uu\delta_s}$, a coefficient linking the elevator's surface deflection to the vehicle's pitching motion significantly affects only the pitch angle (and angle of attack) and the elevator trim angle. The results from varying Z_{uw} and M_{uw} show similar patterns, as expected because these coefficients are related to the same modes of motion as $M_{uu\delta_s}$.

Table 16: Relative changes in trim state values due to variation of $M_{uu\delta_s}$

$M_{uu\delta_s}$ Scale Factor	50%	75%	100%	125%	150%
Angle of attack	28.99%	11.71%	0	-8.37%	-14.63%
Sideslip angle	0.08%	0.03%	0	-0.02%	-0.04%
Roll angle	0.08%	0.03%	0	-0.02%	-0.04%
Pitch angle	28.93%	11.68%	0	-8.36%	-14.60%
Propeller rotation	0.04%	0.01%	0	-0.01%	-0.02%
Prop inflow speed	0.02%	0.01%	0	-0.01%	-0.01%
Elevator angle	-41.95%	-17.70%	0	13.29%	23.56%
Rudder angle	0.08%	0.03%	0	-0.02%	-0.04%
Mechanical torque	0.04%	0.01%	0	-0.01%	-0.02%
# of iterations	**6**	**6**	**7**	**7**	**7**

Table 17: Relative changes in trim state values due to variation of Z_{uw}

Z_{uw} Scale Factor	50%	75%	100%	125%	150%
Angle of attack	-18.12%	-8.43%	0	7.37%	13.85%
Sideslip angle	-0.05%	-0.02%	0	0.02%	0.04%
Roll angle	-0.05%	-0.02%	0	0.02%	0.04%
Pitch angle	-18.08%	-8.42%	0	7.36%	13.82%
Propeller rotation	-0.02%	-0.01%	0	0.01%	0.02%
Prop inflow speed	-0.01%	-0.01%	0	0.00%	0.01%
Elevator angle	-18.16%	-8.45%	0	7.39%	13.87%
Rudder angle	-0.05%	-0.02%	0	0.02%	0.04%
Mechanical torque	-0.02%	-0.01%	0	0.01%	0.02%
# of iterations	***7***	***7***	***7***	***7***	***6***

Table 18: Relative changes in trim state values due to variation of M_{uw}

M_{uw} Scale Factor	50%	75%	100%	125%	150%
Angle of attack	-24.97%	-11.34%	0	9.51%	17.54%
Sideslip angle	-0.06%	-0.03%	0	0.02%	0.05%
Roll angle	-0.07%	-0.03%	0	0.02%	0.05%
Pitch angle	-24.92%	-11.32%	0	9.49%	17.50%
Propeller rotation	-0.03%	-0.01%	0	0.01%	0.02%
Prop inflow speed	-0.02%	-0.01%	0	0.01%	0.01%
Elevator angle	41.20%	18.14%	0	-14.45%	-26.13%
Rudder angle	-0.07%	-0.03%	0	0.03%	0.05%
Mechanical torque	-0.03%	-0.01%	0	0.01%	0.02%
# of iterations	***7***	***7***	***7***	***7***	***6***

The axial drag coefficient $X_{u|u|}$ affects the trim values of the sideslip angle, the roll angle, propeller rotation, propeller inflow speed, rudder angle and the mechanical torque.

Table 19: Relative changes in trim state values due to variation of axial drag $X_{u|u|}$

$X_{u\|u\|}$ Scale Factor	50%	75%	100%	125%	150%
Angle of attack	-0.05%	-0.03%	0	0.04%	0.00%
Sideslip angle	49.62%	24.75%	0	-24.63%	-22.75%
Roll angle	49.88%	24.94%	0	-24.95%	0.00%
Pitch angle	0.03%	0.02%	0	-0.02%	-0.05%
Propeller rotation	29.19%	13.36%	0	-11.77%	0.00%
Prop inflow speed	35.05%	12.99%	0	-8.33%	0.00%
Elevator angle	-0.06%	-0.04%	0	0.05%	0.00%
Rudder angle	49.62%	24.75%	0	-24.63%	18.17%
Mechanical torque	29.12%	13.32%	0	-11.73%	0.00%
# of iterations	**7**	**7**	**7**	**7**	**7**

The coefficient $N_{uu\delta_r}$ links the rudder deflection to the vehicle's yawing motion; thus, the rudder trim angle and the sideslip angle are affected by altering the value of this coefficient.

Table 20: Relative changes in trim state values due to variation of $N_{uu\delta_r}$

$N_{uu\delta_r}$ Scale Factor	50%	75%	100%	125%	150%
Angle of attack	0.00%	0.00%	0	0.00%	0.00%
Sideslip angle	35.86%	15.68%	0	-12.52%	-22.75%
Roll angle	0.00%	0.00%	0	0.00%	0.00%
Pitch angle	0.08%	0.03%	0	-0.03%	-0.05%
Propeller rotation	0.00%	0.00%	0	0.00%	0.00%
Prop inflow speed	0.00%	0.00%	0	0.00%	0.00%
Elevator angle	0.00%	0.00%	0	0.00%	0.00%
Rudder angle	-28.28%	-12.42%	0	9.98%	18.17%
Mechanical torque	0.00%	0.00%	0	0.00%	0.00%
# of iterations	**7**	**7**	**7**	**7**	**7**

On the basis of this analysis, which has considered changes to individual coefficients, the trim-finding algorithm has been shown to be robust (i.e. has found a solution) to 50% variations in five of the hydrodynamic coefficients that are expected to have most effect on trim-state values.

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION
DOCUMENT CONTROL DATA

1. PRIVACY MARKING/CAVEAT (OF DOCUMENT)

2. TITLE

Trim Calculation Methods for a Dynamical Model of the REMUS 100 Autonomous Underwater Vehicle

3. SECURITY CLASSIFICATION (FOR UNCLASSIFIED REPORTS THAT ARE LIMITED RELEASE USE (L) NEXT TO DOCUMENT CLASSIFICATION)

Document	(U)
Title	(U)
Abstract	(U)

4. AUTHOR(S)

Raewyn Hall and Stuart Anstee

5. CORPORATE AUTHOR

DSTO Defence Science and Technology Organisation
PO Box 1500
Edinburgh South Australia 5111 Australia

6a. DSTO NUMBER	6b. AR NUMBER	6c. TYPE OF REPORT	7. DOCUMENT DATE
DSTO-TR-2576	AR-015-046	Technical Report	August 2011

8. FILE NUMBER	9. TASK NUMBER	10. TASK SPONSOR	11. NO. OF PAGES	12. NO. OF REFERENCES
490-6-395	NAV 07/088	DGMD/DGNHM/ COMMHP	45	4

13. DSTO Publications cRepository

http://dspace.dsto.defence.gov.au/dspace/

14. RELEASE AUTHORITY

Chief, Maritime Operations Division

15. SECONDARY RELEASE STATEMENT OF THIS DOCUMENT

Approved for public release

OVERSEAS ENQUIRIES OUTSIDE STATED LIMITATIONS SHOULD BE REFERRED THROUGH DOCUMENT EXCHANGE, PO BOX 1500, EDINBURGH, SA 5111

16. DELIBERATE ANNOUNCEMENT

No Limitations

17. CITATION IN OTHER DOCUMENTS Yes

18. DSTO RESEARCH LIBRARY THESAURUS http://web-vic.dsto.defence.gov.au/workareas/library/resources/dsto_thesaurus.shtml

Autonomous Vehicles, Control Systems, Stability, Dynamic Characteristics

19. ABSTRACT

The calculations given in this work demonstrate that the trimmed state for a dynamical model of the REMUS 100 autonomous underwater vehicle is readily found using a numerical zero-finding procedure based on Newton-Raphson iteration. This work also presents approximate analytical expressions for the trimmed state that can be used as a starting point for the numerical procedure. The procedure should be applicable to a range of hydrodynamic parameters corresponding to other configurations of the REMUS 100 vehicle and to similar vehicles from other vendors.